

Customer Engineering
Manual of Instruction

7070

Data Processing System
Volume II

© April 1, 1960, by International Business Machines Corporation
Printed in U.S.A.
Form 223-6870-0

IBM[®] Customer Engineering

Manual of Instruction

7601 A & P Control Unit

7601 Arithmetic and Program
Control Unit

IBM ARITHMETIC AND PROGRAM CONTROL UNIT

CONTENTS (VOLUME I)

1.0.00	GENERAL INFORMATION.....	1	2.3.04	Validity Check.....	24
1.1.00	Introduction.....	1	2.3.05	Match Circuit.....	26
1.2.00	General Machine Logic.....	1	2.3.06	Field Ring.....	27
1.3.00	Physical Layout.....	1	2.3.07	Field Length Register.....	28
1.4.00	Machine Language.....	1			
1.5.00	Input-Output (I/O).....	1			
1.6.00	Control.....	1			
			3.0.00	INSTRUCTION PROCESSING ELEMENTS.....	29
2.0.00	DATA PROCESSING ELEMENTS.....	4	3.1.00	Clocking.....	29
2.1.00	Register Unit Theory.....	4	3.1.01	Basic System.....	29
2.1.01	Accumulator 1.....	6	3.1.02	Basic Circuit and Timing.....	30
2.1.02	Accumulator 2.....	7	3.2.00	Control Rings.....	33
2.1.03	Accumulator 3.....	8	3.2.01	Program and Program Cycle Rings.....	33
2.1.04	Auxiliary Register.....	9	3.2.02	Cycle Control Rings.....	34
2.1.05	Arithmetic Register.....	10	3.3.00	Register Units.....	34
2.1.06	Word Buffer Register.....	11	3.3.01	Latch Registers.....	34
2.1.07	Address Start Register.....	12	3.3.02	Core Registers.....	34
2.1.08	Address Control Register A.....	13			
2.2.00	Transmission Lines.....	13	4.0.00	PROGRAM EXECUTION.....	43
2.2.01	Arithmetic Bus (AB).....	14	4.1.00	Instruction Cycle.....	44
2.2.02	Information Bus (IB).....	14	4.1.01	Introduction.....	44
2.2.03	Computer Address Bus (CAB).....	15	4.1.02	Data Flow and Circuits.....	44
2.2.04	Adder Output Channel.....	16	4.2.00	Index Cycle.....	51
2.2.05	Channel 1 (Adder Entry B).....	16	4.2.01	Introduction.....	51
2.2.06	Channel 2 (Adder Entry B).....	17	4.2.02	Data Flow and Circuits.....	51
2.3.00	Processing Units.....	17	4.3.00	Data Cycle.....	55
2.3.01	Adder.....	18	4.3.01	Introduction.....	55
2.3.02	Significant Digit Scanner.....	21	4.3.02	Circuits.....	56
2.3.03	Shift Register.....	22	4.4.00	Field Control.....	58

CONTENTS (VOLUME II)

5.0.00	INSTRUCTION EXECUTION.....	61	5.8.08	Branch High, Low or Equal.....	322
5.1.00	Add-Subtract Store.....	61	5.8.09	Field Overflow Control.....	323
5.1.01	Add-Subtract Accumulator #		5.8.10	Branch on Alteration Switch or Channel Busy.....	325
	Add-Subtract Absolute Accumulator 1.....	65	5.9.00	Indexing Instructions.....	328
5.1.02	Zero and Add-Subtract Accumulator #		5.9.01	Branch Compared Index Word.....	328
	Zero and Add-Subtract Accumulator 1.....	82	5.9.02	Branch if Index Word Indexing Portion is Non-Zero.....	330
5.1.03	Add To Storage From Accumulator #		5.9.03	Branch if Indexing Portion is Minus.....	334
	Subtract Accumulator # From Storage		5.9.04	Index Word Load.....	334
	Add to Absolute Storage From Accumulator #.....	97	5.9.05	Index Word Unload.....	335
5.1.04	Zero Storage and Store Accumulator #		5.9.06	Index Word Zero and Add to-Subtract from Index Portion.....	335
	Store Accumulator #		5.9.07	Index Word Add to-Subtract From Indexing Portion (XA).....	339
	Store Digit From Accumulator # and		5.9.08	Index Word Set Non-Indexing Portion.....	347
	Ignore Sign (Store Absolute).....	115	5.9.09	Index Word Load and Interchange.....	347
5.2.00	Multiply (+53) Introduction.....	132	5.9.10	Branch Incremented Index Word.....	347
5.2.01	(+53) Multiply.....	135	5.9.11	Branch Decrement Index Word.....	347
5.3.00	Divide (-53) - Introduction.....	158	5.10.00	Table Lookup.....	360
5.3.01	Divide.....	160	5.10-1	TLU Equal, and Equal or High.....	367
5.4.00	Shift Operations.....	181	5.10-2	TLU Lowest.....	375
5.4.01	+50 Shift Single.....	197	5.11.00	Priority Processing.....	380
5.4.02	-50 Double Shift - Full Factor.....	211	5.11.01	Priority Circuit Operation.....	380
5.4.03	-50 Shift Double - Shift From Point.....	233	5.11.02	Priority Codes.....	393
5.5.00	Block Transfer Operations.....	255	5.12.00	Card Control Operations.....	403
5.5.01	Record Scatter and Record Gather.....	255	5.12.01	Unit Record Signal.....	403
5.5.02	Edit Numerical to Alphameric.....	261	5.12.02	Read Operation.....	404
5.5.03	Edit Numerical to Alphameric with Sign Control.....	271	5.12.03	Write Operation.....	415
5.5.04	Edit Numerical to Alphameric with Blank Insertion.....	271	5.12.04	Write or Punch Invalid Operation.....	423
5.5.05	Edit Alphameric to Numerical.....	280	5.12.05	Type Operation.....	426
5.6.00	Compare Instructions.....	286	5.13.00	Console Operations.....	436
5.6.01	Compare Accumulator to Storage.....	286	5.13.01	Display on Stop.....	436
5.6.02	Compare Absolute.....	294	5.13.02	Display on Console Keyed Address.....	441
5.6.03	Compare Storage to Digit.....	294	5.13.03	Store In Memory from Console.....	444
5.7.00	Sign Control, Code - 03.....	297	5.14.00	Tape Channel Operations.....	447
5.8.00	Branch Instructions.....	306	5.14.01	Tape Control Operation.....	447
5.8.01	Halt and Proceed, and Halt and Branch.....	307	5.14.02	Disk Seek Operation.....	454
5.8.02	Branch and No Op.....	309	5.14.03	Disk Read/Write Operation.....	461
5.8.03	Branch and Load Location in Index Word.....	311	5.14.04	Disk Arm Release (-95).....	467
5.8.04	Electronic Switch Control, OP +61, +62, +63.....	313	5.15.00	Start, Stop, and Reset Operations.....	474
5.8.05	Branch If Zero in Accumulator.....	318	5.15.01	Start Controls.....	474
5.8.06	Branch If Minus in Accumulator.....	318	5.15.02	Stop Controls.....	474
5.8.07	Branch If Overflow in Accumulator #.....	321	5.15.03	Reset Controls.....	483

FOREWORD

This manual is intended for use in training Customer Engineers in the operation of the IBM 7601 Arithmetic & Program Control Unit used with the IBM 7070 System. Illustrations accompany the text. In most cases, the sections, sub-sections, and items contained within this manual, are written so that the preceding information serves as a building block for subsequent study.

Sections One through Four inclusive are contained in Volume I while Section Five is contained in Volume II.

Each program controlled operation is discussed in detail and is accompanied by operation flow charts and sequence charts. Major signals, key timings, and logic locations are also given in chart form. Operations under CE Panel control are discussed in a like manner.

Future engineering changes may cause minor discrepancies in some of the timings given in sequence charts and other charts (signal, timing and location). They should not change the philosophy of the operations unless the changes are of a major nature and constitute revisions to data flow and operation sequence.

ILLUSTRATIONS (VOLUME I)

Figure	Title	Page	Figure	Title	Page
1.2-1	A&P Data Flow.....	2	3.1-1	7070 Clocking System.....	29
1.3-1	A&P Physical Location Chart.....	3	3.1-2A	A&P 4 Usec Clock.....	31
			3.1-2B	A&P 4 Usec Clock Timing.....	32
2.1-1	Serial Transfer.....	4	3.2-1	Program Ring Timing.....	33
2.1-2	Parallel Transfer.....	5	3.2-2	Program Ring and Controls.....	35
2.1-3	Extender.....	6	3.2-3	Control Ring Timing.....	36
2.1-4	Accumulator 1.....	7	3.2-4	Store Control Ring Timing.....	37
2.1-5	Accumulator 2.....	7	3.3-1	Program Register Latches.....	38
2.1-6	Accumulator 3.....	8	3.3-2	Program Register D.....	39
2.1-7	Auxiliary Register.....	9	3.3-3	Program Register Controls.....	40
2.1-8	Arithmetic Register.....	10	3.3-4	Instruction Counter.....	41
2.1-9	Word Buffer Register.....	11	3.3-5	One-Up Adder.....	42
2.1-10	Address Start Register.....	13			
2.1-11	Address Control Register A.....	13			
2.3-1	Single Plane Adder.....	18			
2.3-2	Read-In Windings - Entry A and Entry B.....	19	4.0-1	Program Execution Data Flow.....	43
2.3-3	Hysteresis Loop.....	20	4.1-1	Program Controls for Initial Access Codes.....	45
2.3-4	Adder Timing.....	20	4.1-2	Control Rings.....	46
2.3-5	Significant Digit Scanner.....	21	4.1-3	Instruction Cycle Sequence.....	47
2.3-6	Shift Register.....	23	4.1-4	Instruction Counter Control.....	50
2.3-7	Validity Check Circuit.....	25	4.2-1	Index Cycle Sequence.....	53
2.3-8	A&P Match Circuit.....	26	4.2-2	Index Operation Data Flow and Control Circuits.....	54
2.3-9	Field Ring.....	27	4.3-1	Data Cycle Sequence.....	56

ILLUSTRATIONS (VOLUME II)

5.1-1	Add To Accumulator Data Flow.....	61	5.4-12	Split Shift Set Up.....	192
5.1-2	Add To Storage and Store Data Flow.....	63	5.4-13	Shift Right A - Point Stays Within A1.....	193
5.1-3	Add - Subtract Sign Control.....	64	5.4-14	Shift Right A1 - Point Shifts Out of A1.....	193
5.1-4	Add To Accumulator.....	84	5.4-15	Shift Left A1 - Point Stays Within A1.....	194
5.1-5	Add To Accumulator.....	85	5.4-16	Shift Left A1 - Point Shifts Out of A1.....	194
5.1-6	Zero and Add.....	95	5.4-17	Shift Right A2 - Point Stays Within A2.....	195
5.1-7	Zero and Add.....	96	5.4-18	Shift Right A2 - Point Shifts Out of A2.....	195
5.1-8	Add To Storage.....	98	5.4-19	Shift Left A2 - Point Stays Within A2.....	196
5.1-9	Add To Storage.....	99	5.4-20	Shift Left A2 - Point Shifts Out of A2.....	197
5.1-10	Store.....	129	5.4-21	Shift 1 Cycle Setup - Right Single, Left Single.....	200
5.1-11	Store.....	130	5.4-22	Shift Right Round Single.....	203
5.1-12	Store.....	131	5.4-23	Shift Left and Count Single.....	206
5.2-1	Multiplicand Factor Data Flow.....	133	5.4-24	Shift Left Count - Count Storage Cycle Single and Double.....	207
5.2-2	Multiply Add Cycle Data Flow.....	134	5.4-25	Shift Right Double.....	213
5.2-3	Zero and Add To A2 - Multiply Operation.....	137	5.4-26	Shift Left Double.....	216
5.2-4	Form X2 Multiple.....	139	5.4-27A	Shift Right Round Double.....	220
5.2-5	Form X4 Multiple.....	140	5.4-27B	Shift Right Round Double - 10 Positions.....	221
5.2-6	X4 Multiple Control.....	143	5.4-28A	Shift Left and Count Double (A1 Non Zero).....	226
5.2-7	X2 Multiple Control.....	146	5.4-28B	Shift Left and Count Double (A1 Zero).....	227
5.2-8	Multiply Idle - X1 Multiple Control.....	149	5.4-28C	Shift Left and Count Double (A1 & A2 Zero).....	229
5.2-9	Multiply Idle - Multiply End.....	154	5.4-29	Shift From Point Basic Setup.....	235
5.2-10	Product Development During Multiplication.....	155	5.4-30	Shift Right Point in A1.....	238
5.2-11A	Major Timing For Product Development.....	156	5.4-31	Shift Right Point in A2.....	243
5.2-11B	Major Timing For Product Development.....	157	5.4-32	Shift Left in A1.....	247
5.3-1	Dividend Complement Data Flow.....	158	5.4-33	Shift Left Point in A2.....	252
5.3-2	Dividend Reduction Data Flow.....	159	5.5-1	Record Gather Block Diagram.....	256
5.3-3	Zero and Add To A3 - Divide Operation.....	162	5.5-2	Block Transmission - Record Gather.....	257
5.3-4	Complement A2 and A1.....	164	5.5-3	Data Flow for Transfer Control 1 Operations.....	259
5.3-5	Start Divide.....	168	5.5-4	Data Flow for Transfer Control 2 Operations.....	259
5.3-6	Continue Divide Add.....	171	5.5-5	Data Flow for Transfer Control 3 Operations - Gather Only.....	260
5.3-7	Divide Left Shift Add.....	174	5.5-6A	Block Transfer Circuits.....	262
5.3-8	Divide End.....	178	5.5-6B	Block Transfer Circuits.....	263
5.3-9	Division Problem.....	180	5.5-7A	Sequence for Block Transfer.....	264
5.3-10A	Timing Chart For Division Problem.....	181	5.5-7B	Sequence for Block Transfer.....	265
5.3-10B	Timing Chart For Division Problem.....	181	5.5-8	Block Transmission - Record Scatter.....	266
5.4-1	Single Shift Right or Left.....	183	5.5-9	Data Flow for Transfer Control 3 Operations - Scatter Only.....	267
5.4-2	Shift Right and Round.....	184	5.5-10	Edit Numerical to Alphameric Flow Chart.....	269
5.4-3	Shift Left and Count (Cycle 1).....	184	5.5-11A	Edit Numerical to Alphameric Logic.....	272
5.4-4	Shift Count Store (Cycle 2).....	185	5.5-11B	Edit Numerical to Alphameric Logic.....	273
5.4-5	Double Shift Right - Less Than Ten.....	187	5.5-11C	Edit Numerical to Alphameric Logic.....	274
5.4-6	Double Shift Right - Ten or Greater.....	187	5.5-12A	Sequence for Edit Numerical to Alphameric.....	275
5.4-7	Double Shift Left - Less Than Ten.....	188			
5.4-8	Double Shift Left - Ten or Greater.....	188			
5.4-10	Round Cycle - Double Right Shift.....	189			
5.4-11	Double Shift Left and Count - A1 Zero.....	190			

Figure	Title	Page	Figure	Title	Page
5.5-12B	Sequence for Edit Numerical to Alphameric.....	276	5.9-18	Seq. for Index Word Load and Interchange OP48.....	351
5.5-13	Logic for Edit Numerical to Alphameric with Sign Control - Op NASC.....	277	5.9-19	Branch Incremented Index Word, Op + 49.....	352
5.5-14	Logic for Edit Numerical to Alphameric with Blank Insertion - Op NABI.....	278	5.9-20	Sequence for Branch Incremented Index Word, Op +49.....	353
5.5-15	Sequence for Numerical to Alphameric Conversion with Blank Insertion/Sign Control.....	279	5.9-21	Branch Decrement Index Word Block Diagram.....	356
5.5-16	Edit Alpha to Numeric Flow Chart.....	281	5.9-22	Branch Decrement Index Word, Op - 49.....	358
5.5-17A	Logic for Edit Alphameric to Numerical.....	282	5.9-23	Sequence for Branch Decrement Index Word, Comp Add, No Sign Change, No Comp Adjust.....	359
5.5-17B	Logic for Edit Alphameric to Numerical.....	283	5.10-1	TLU Equal Operation Flow Chart.....	362
5.5-17C	Logic for Edit Alphameric to Numerical.....	284	5.10-2	TLU Equal or High Operation Flow Chart.....	363
5.5-18	Sequence for Edit Alphameric to Numerical.....	285	5.10-3	TLU Lowest Operation Flow Chart.....	364
5.6-1	Compare Accumulator Block Diagram.....	287	5.10-4A	TLU Equal Logic.....	366
5.6-2A	Compare Accumulator and Compare Absolute.....	289	5.10-4B	TLU Equal Logic.....	367
5.6-2B	Compare Accumulator and Compare Absolute.....	290	5.10-5	Additional Logic for Op TLU Lowest.....	368
5.6-3	Switching Conditions for Compare Accumulator.....	292	5.10-6A	TLU Sequence of Operation.....	372
5.6-4	Sequence for Compare Accumulator to Storage, Op + 15, 25, 35, Signs Not Equal.....	292	5.10-6B	TLU Sequence of Operation.....	373
5.6-5	Sequence for Compare Accumulator to Storage, Op + 15, 25, 35, Field Equal and Same Signs.....	293	5.10-6C	TLU Sequence of Operation.....	374
5.6-6	Compare Storage to Digit.....	296	5.10-7A	TLU Lowest Sequence of Operation.....	378
5.6-7	Compare Storage to Digit Sequence.....	297	5.10-7B	TLU Lowest Sequence of Operation.....	379
5.7-1A	Sign Control.....	299	5.11-1	Flow Chart for Priority Circuit Operation.....	381
5.7-1B	Sign Control.....	300	5.11-2A	Logic for Interrupt Operation (Card or Inq).....	382
5.7-2	Sign Control, Compare (-03 00X0 XXXX).....	301	5.11-2B	Logic for Interrupt Operation (Card or Inq).....	383
5.7-3	Sign Control, Set (-03 00X1 XXXX).....	303	5.11-3	Sequence of Interrupt Operation (Card Unit).....	385
5.7-4	Sign Control, Set Sign Change Switch (-03 00 02 XXXX Set to Sense) (-03 00 03 XXXX Set to Stop).....	304	5.11-4	Logic for Interrupt Operation (Tape Unit).....	386
5.7-5	Sign Control, Branch (-03 0004 XXXX) Sign Store Latch Turned On.....	305	5.11-5	Sequence for Interrupt Operation (Tape Unit).....	387
5.8-1	Halt and Branch, Halt and Proceed.....	306	5.11-6	Interrupt Core Storage Locations.....	388
5.8-2	Halt and Branch.....	307	5.11-7	Logic for Stacking Latch, Op - 61.....	391
5.8-3	Halt and Proceed.....	308	5.11-8	Logic for Stacking Latch Reset, Op - 62.....	392
5.8-4	Branch and No Op.....	310	5.11-9	Sequence for Stacking Latch Set and Reset, Op Codes - 61 and - 62.....	393
5.8-5	Sequence for Branch (B).....	310	5.11-10	Logic for Priority Control, Op + 55.....	394
5.8-6	Branch and Load Location in Index Word.....	312	5.11-11	Sequence for Priority Control, Op + 55.....	395
5.8-7	Branch and Load Location in Index Word.....	313	5.11-12	Logic for Stacking Latch Test, Op + 60.....	397
5.8-8	Electronic Switch Control.....	316	5.11-13	Sequence for Stacking Latch Test, Op + 60 (No Indexing).....	398
5.8-9	Sequence for Electronic Switch Control.....	317	5.11-14	Flow Chart for Priority Release, Op + 64.....	399
5.8-10	Branch If Zero In Accumulator 1.....	319	5.11-15	Logic for Priority Release, Op + 64.....	400
5.8-12	Sequence for Branch if Zero Accumulator, Op + 10.....	320	5.11-16	Sequence for Priority Release, Op + 64.....	401
5.8-13	Sequence for Branch Minus.....	321	5.12-1	Logic for Set Program Exit Switch.....	405
5.8-14	Branch On Accumulator Overflow.....	322	5.12-2	Read Operation Block Diagram.....	406
5.8-15	Branch On High, Low, Equal.....	323	5.12-3A	Logic for Card Control Read Op.....	408
5.8-16	Field Overflow Control.....	325	5.12-3B	Logic for Card Control Read Op.....	409
5.8-17	Alteration Switch, Channel Busy, Branch.....	327	5.12-3C	Logic for Card Control Read Op.....	411
5.9-1	Branch Compared Index Word, Op - 43 (Tran IP Less or Equal FP).....	330	5.12-4	Sequence for Card Control Read Op.....	417
5.9-2	Branch Compared Index Word, Op - 43.....	331	5.12-5	Drum to WBR Transfer.....	418
5.9-3	Branch If Index Word Indexing Portion is Non-Zero (+44 TRNZ) Branch If Index Word is Minus (-44 TRM).....	332	5.12-6	End of File Sequence.....	419
5.9-4	Sequence for Branch If Index Word is Non-Zero, Op +44.....	333	5.12-7	Write Operation Block Diagram.....	424
5.9-5	Sequence for Branch If Index Word is Minus, Op-44.....	334	5.12-8A	Logic for Card Control Write Op.....	425
5.9-6	Index Word Load, Op + 45.....	336	5.12-8B	Logic for Card Control Write Op.....	427
5.9-7	Sequence for Index Word Load, Op +45.....	337	5.12-8C	Sequence for Card Control Write Op.....	430
5.9-8	Sequence for Index Word Unload, Op - 45.....	337	5.12-9	Sequence for Type.....	431
5.9-9	Index Word Unload, Op -45.....	338	5.12-10A	Sequence for Type.....	432
5.9-10	Index Word Zero and Add to - Subtract from Indexing Portion, Op $\pm$ 46.....	340	5.12-10B	Logic for Card Control Type Op.....	433
5.9-11	Sequence for Index Word Zero and Add to - Subtract from Indexing Portion, Op $\pm$ 46.....	341	5.12-11	Sequence for Error in 1st RCW.....	434
5.9-12	Index Word Add to - Subtract from Indexing Portion, Op $\pm$ 47.....	343	5.12-12	Sequence for RCW Error Operation.....	435
5.9-13	Sequence for Index Word Add to - Subtract from Indexing Portion, Op $\pm$ 47.....	344	5.12-13A	Sequence for RCW Error Operation.....	436
5.9-14	Sequence for Index Word Add to - Subtract from Indexing Portion, Complement Add with Recomplement.....	345	5.12-13B	Sequence for RCW Error Operation.....	437
5.9-15	Index Word Set Non-Indexing Portion.....	348	5.13-1	Display on Stop Operation.....	437
5.9-16	Sequence for Index Word Set Non-Indexing Portion, Op + 48.....	349	5.13-2	Logic for Display on Stop.....	438
5.9-17	Index Word Load and Interchange, Op - 48.....	350	5.13-3A	Sequence for Display on Stop.....	439
			5.13-3B	Sequence for Display on Stop.....	442
			5.13-4	Logic for Display on Console Keyed Address.....	443
			5.13-5	Sequence for Display on Console Keyed Address.....	444
			5.13-6	Logic for Store In Memory from Console.....	445
			5.13-7	Sequence for Store In Memory from Console.....	449
			5.14-1	Tape Control Op Flow Chart.....	450
			5.14-2	Tape Op $\pm$ 8X.....	456
			5.14-3	DS Seek Operation Flow Chart.....	459
			5.14-4	Disk Seek Operation.....	462
			5.14-5	$\pm$ 9X DS Read/Write Flow Chart.....	463
			5.14-6	$\pm$ 9X DS R/W.....	470
			5.14-7	- 95 Disk Arm Release Flow Chart.....	471
			5.14-8	Disk Arm Release.....	481
			5.15-1	Arithmetic Register Zero Insert Error Check.....	483
			5.15-2	Simplified Logic of Reset Switching.....	485
			5.15-3	Controlled Computer Reset Timing Chart.....	

1.0.00 GENERAL INFORMATION

1.1.00 INTRODUCTION

The Arithmetic and Program Control unit (A & P) is the center of activity in the IBM 7070 System. It contains the circuits that perform arithmetic, logical, and other programming operations.

The instructions of the program control the A & P unit. The logic of the A & P circuits control how the information is processed to perform an instruction.

1.2.00 GENERAL MACHINE LOGIC

Figure 1.2-1 shows the paths that information can follow as it goes from one point to another in the A & P unit. The instruction being operated on by the A & P unit controls when and where to the information will be moved.

1.3.00 PHYSICAL LAYOUT

The 7070 A & P unit occupies two SMS Sliding Gate Modules called A & P Frame 11 and A & P Frame 12. Figure 1.3-1 gives the frame, gate and panel location of the various A & P registers, rings, etc.

1.4.00 MACHINE LANGUAGE

Instructions and data used by the A & P unit must be in the two-out-of-five fixed-count code. Inputs to the 7070 system can be in many forms but, information must be decoded before the A & P unit can use it.

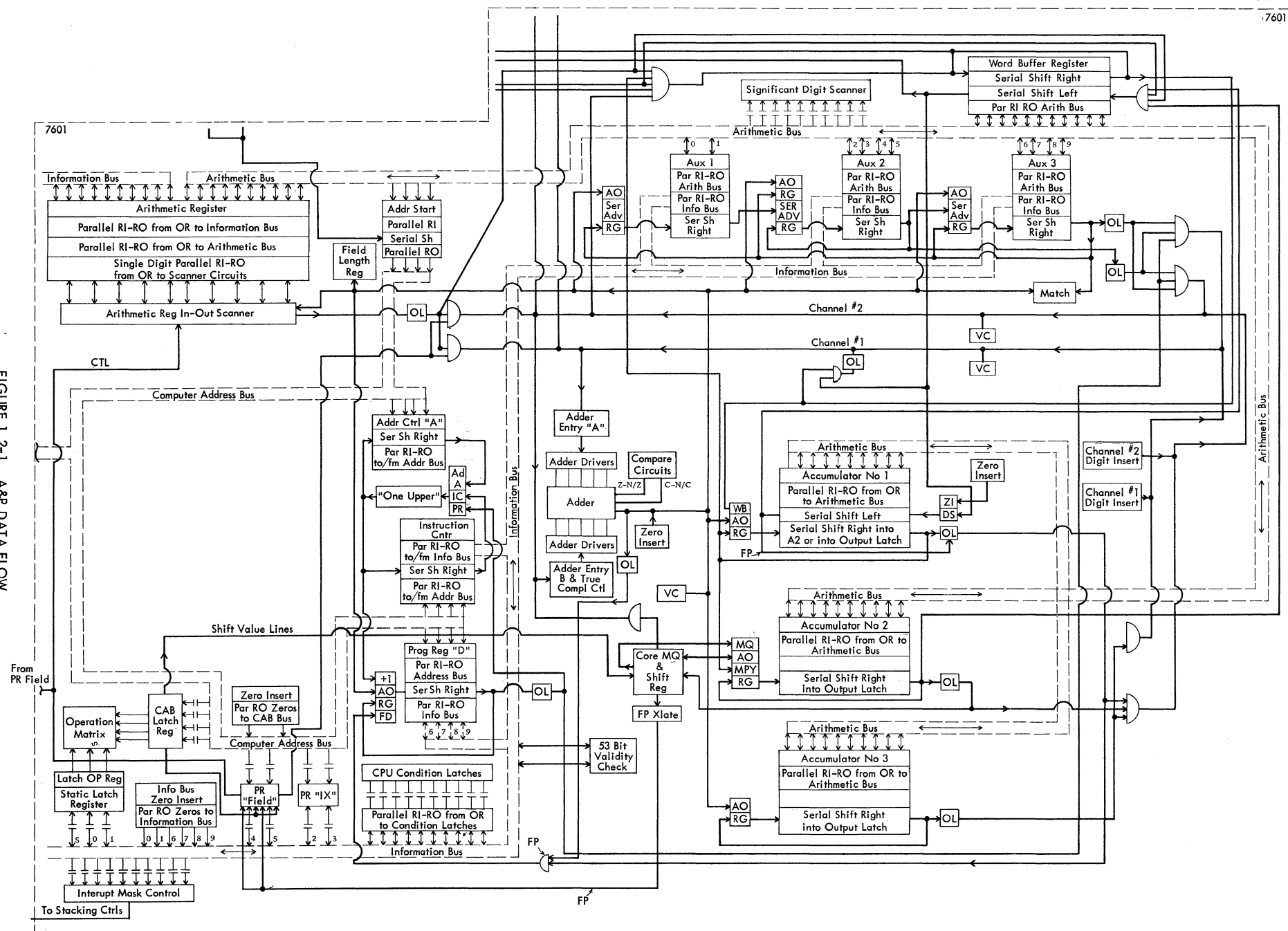
1.5.00 INPUT-OUTPUT(I/O)

The direct data input and output for the 7070 system is core storage. Information can be read from cards, tape, disk storage, and also keyed in from a typewriter. Information can be taken out of the system on to cards, tape, disk storage, or a printed form. However, as data passes between I/O and the A & P unit, it must pass through core storage.

The 7070 system has only one core storage unit and it is time-shared by the I/O system and the A & P unit.

1.6.00 CONTROL

The stored program controls the 7070 system. Program instructions are brought into the A & P unit, one at a time, and are directed to the program register. From there the instruction is decoded to direct operation of the units concerned.



ARITHMETIC AND PROGRAM CONTROL UNIT			
UNIT	FRAME, GATE AND PANEL	UNIT	FRAME, GATE AND PANEL
DATA REGISTERS		PRG. REG. DATA	11A 4
A 1	11B 1	PRG. REG. D CTRLS	12A 1
A 1 CTRLS	11D 3	PRG. REG. OP.	11C 1
A 2	11A 1	PRG. REG. IX.	11C 1
A 2 CTRLS	11D 3	PRG. REG. FLD.	11C 1
A 3	11A 1	CAB LATCH REG.	11A 3
A 3 CTRLS	11D 1 + 11D 3	ADDER	11B 4
AUXR 1	11A 1	ADDER ENTRY A	11B 4
AUXR 1 CTRLS	11C 2	ADDER ENTRY B	11B 4
AUXR 2	11A 2	ADDER OUTPUTS (C/NC, Z/NZ)	11B 4
AUXR 2 CTRLS	11C 2	ADDER TRUE-COMPLEMENT	11D 4 + 12B 2
AUXR 3	11A 2		
AUXR 3 CTRLS	11C 2	TRANSFER BUSES	
ARITHMETIC REGISTER	11A 2	INFORMATION BUSS (CAP)	CORE STORAGE
ARITH. REG. CTRLS & SCANNER	11A 2 + 12A 1	I.B. SENSING	11A 4
FIELD RING	12D 1	I.B. CTRLS	11D 2
FLD. LENGTH REG.	11C 1	I.B. INSERTS	11C 4
WORD BUFFER	11B 1	I.B. VALIDITY	11A 4
WRD. BUF. CTRLS	12A 3	ARITHMETIC BUSS (CAP)	11B 2
		A.B. CTRLS	11B 2
CLOCKING		CAB BUS	11A 3
PROGRAM RING	12D 1	CAB INSERTS	11A 3
PRG. RING CTRLS	11D 1 + 12D 1	CAB CTRLS	11A 3
INSTRUCTION CTRL RING	12C 1	CHANNEL 1 MIX	11B 4
I. CTRL RING CTRLS	11D 2	CHAN. 1 VALIDITY	11B 4
INDEX CTRL RING	12C 1	CHAN. 1 INSERTS	11D 4
IX. CTRL RING CTRLS	11D 2	CHANNEL 2 MIX	11B 4
DATA CTRL RING	12C 1	CHAN. 2 VALIDITY	11B 4
D. CTRL RING CTRLS	11D 2	CHAN. 2 INSERTS	11D 4
STORE CTRL RING	12C 1	ADDER OUTPUT CHANNEL (CAP)	11B 4
STR. CTRL RING CTRLS	11D 2	A.O.C. VALIDITY	11B 4
MIP & MOP	11D 2		
BASIC 4 US CLOCK		PROGRAM CTRLS	
FRAME 11 GATES A & B	11B 3	PRESENSE OP. CODE	11C 1
FRAME 11 GATES C & D	11D 1	OP. MATRIX	11C 3 + 12B 1
FRAME 12 GATES A & B	12B 3	ARITHMETIC CODES	11C 3
FRAME 12 GATES C & D	12D 1	SHIFT CODES	12B 1
		BLOCK TRANSFER	12A 2
PROCESS REGISTERS		TABLE LOOK UP	12A 2
SIGNIFICANT DIGIT SCANNER	11B 2	CARD CONTROLS	12C 4
S.D.S. CTRLS	12B 2	MULTIPLY CONTROLS	12B 3
SHIFT REGISTER	11B 3	DIVIDE CONTROLS	12B 1
SH. REG. CTRLS	12B 4	INTERRUPT CONTROLS	12B 3
ADDRESS START	11A 1	MASK	11C 2
ADDR. STT. CTRLS	12A 1	TAPE CONTROL	12B 3
ADDRESS CTRL A	11A 3	ERROR CONTROLS	12C 2
ADDR. CTRL A CTRLS	11A 3	RESET CONTROLS	12C 1
INSTRUCTION COUNTER	11A 4	MANUAL STORE & DISPLAY	12C 4
I.C. CTRLS	12A 4	CONDITION LATCHES	11D 4
I.C. ONE UPPER	11B 3	SIGN CONTROL	12B 2
I.C. COMPUTER MATCH	11B 3	INDEX OP. CODES	11C 3

FIGURE 1.3-1. A&P PHYSICAL LOCATION CHART

2.0.00 DATA PROCESSING ELEMENTS

2.1.00 REGISTER UNIT THEORY

The magnetic-core shift-register system is used extensively for data processing registers in the A & P area. Data is normally parallel-transferred between registers for location and serially-transferred for most processing. The system lends itself to conversions to and from CTDL Logic.

A basic register is composed of shift register cards (SR 1 or SR 3) with various configurations of driver elements (Figures 2.1 - 1 and 2). The cores are all reset at A-pulse of each clock cycle by Read-Out Drivers. Sensing of a stored bit being reset is under control of the Read-Out Control Driver, which sets the bias for the transistor amplifiers. The amplifiers provide the power required to store the bit as a charge on a capacitor. The capacitor charge is used to set a magnetic core at C pulse time by gating with a Read-In Driver. The bit may be read back into the same core as a re-generation or it may read into another core as either a serial or parallel transfer. The multiple windings on the magnetic cores permit two read-outs or transfers simultaneously.

During the time when a bit is stored as a capacitor charge, the line can be sampled with Capacitor Sense Amplifiers or CTDL Emitter Followers. Sampling of this type permits the operation of validity checking circuits or other logic and is limited to a discharge value which still permits setting a core. Sometimes, it is only necessary that other logic be operated without setting a core. An external capacitor card is used for this purpose. It has a dummy load circuit to allow dumping the remaining capacitor charge.

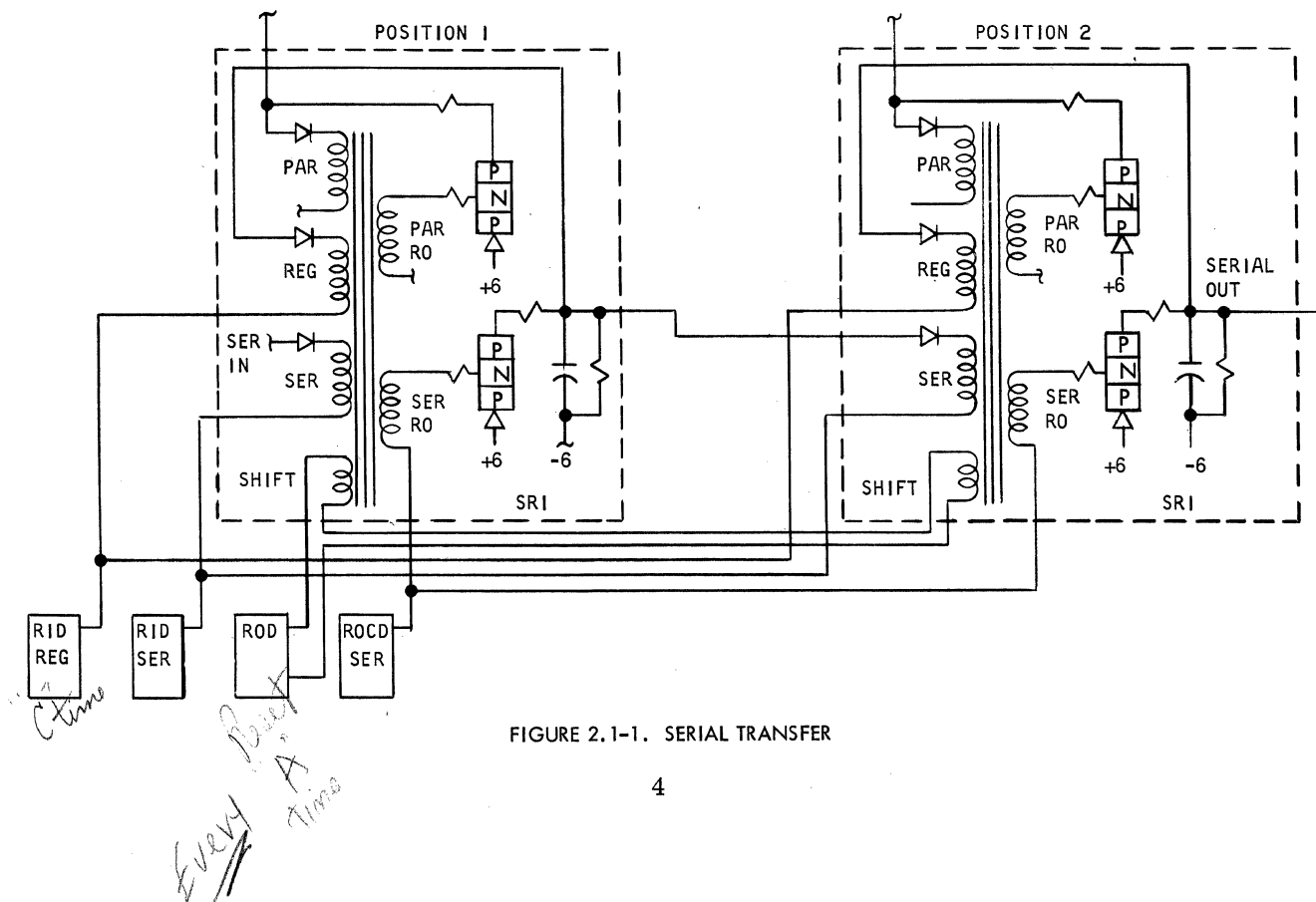


FIGURE 2.1-1. SERIAL TRANSFER

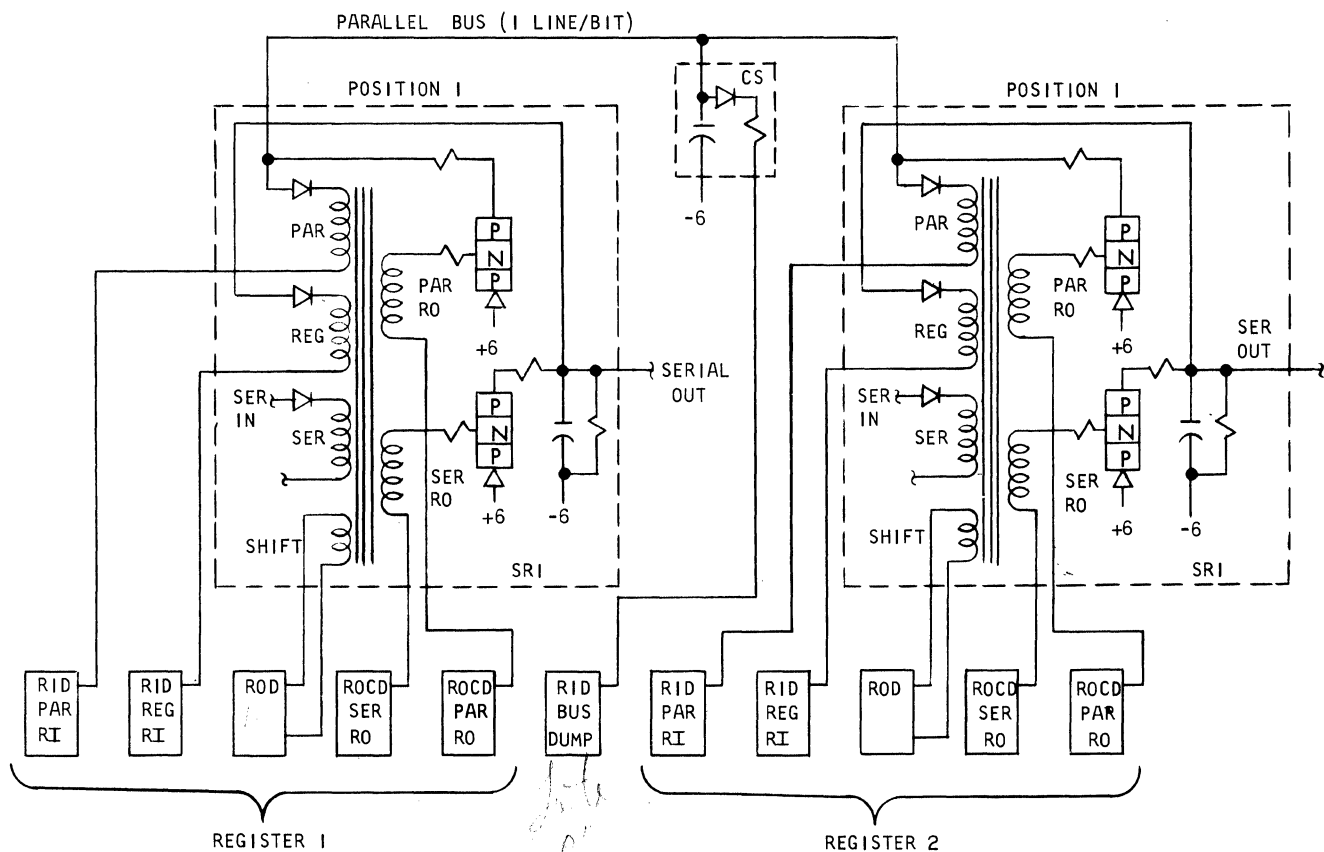


FIGURE 2.1-2. PARALLEL TRANSFER

Data or logic can be entered into a register in two ways depending on the control problem. A core may be set directly with a current pulse at C-time to a read-in winding. The core may be set indirectly by charging the transfer capacitor at A-time and setting the core regeneration read-in at C-time. In either charging the capacitor or setting a core only one drive circuit is used at a time to prevent invalid characters. In some cases, this requires separate drivers either for read-out or read-in. A bit insert driver can be used to drive either a core or a capacitor. Expansion of the number of core inputs can be made with an extender card (SR2 or SR4, Figure 2.1-3). The extender functions like a separate register for one cycle. The data bit is stored in the separate core with its read-out circuits feeding the associated register position capacitors. The bit is regenerated only in the register core leaving the extender blank until the next entry.

Transfer between registers is explained further under Transmission Lines in Section 2.2.00.

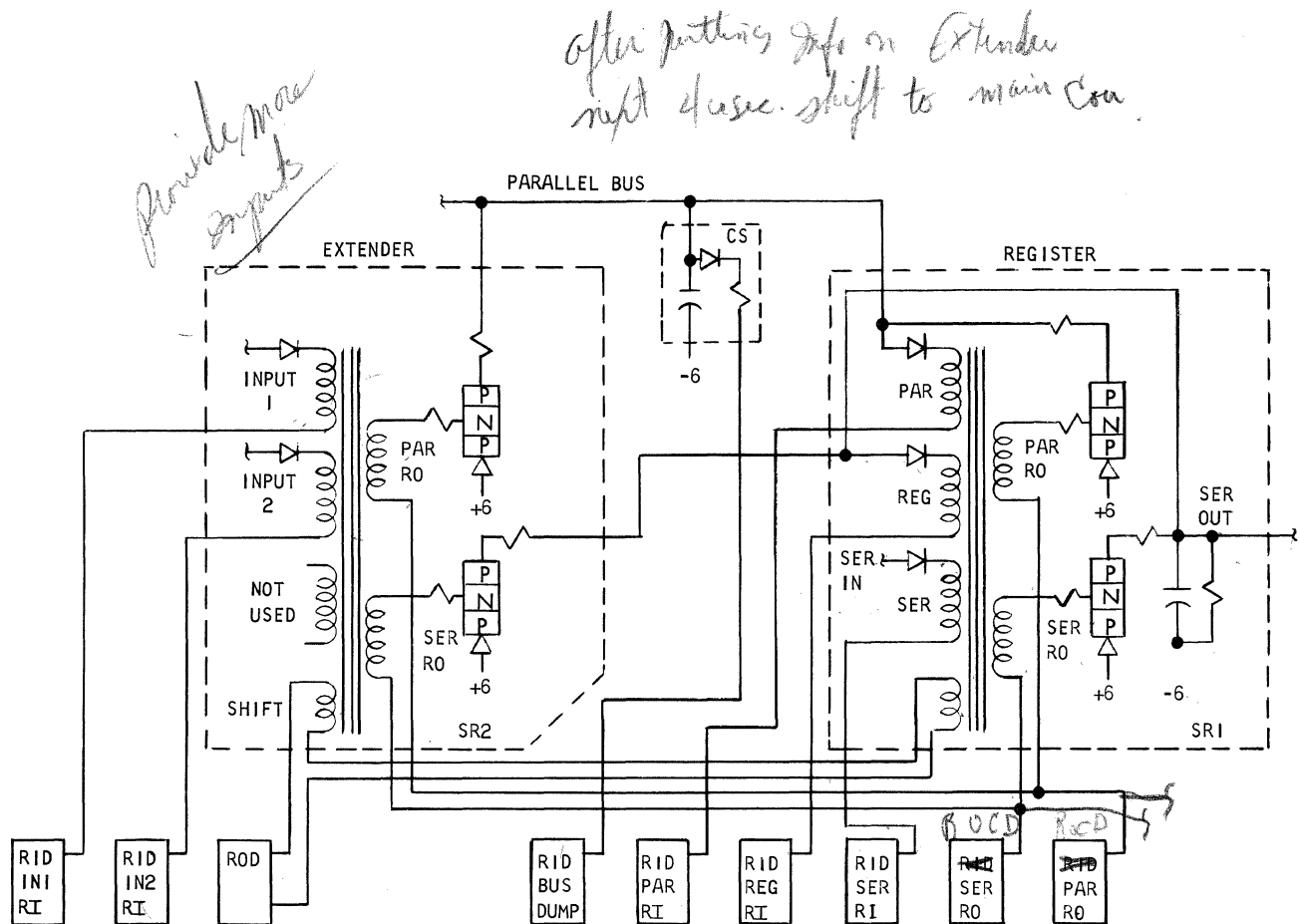


FIGURE 2.1-3. EXTENDER

2.1.01 ACCUMULATOR 1 (Figure 2.1-4)

Accumulator 1 is a magnetic-core shift register with ten digit positions and a sign position. The register can be used as either an operated or addressed location in normal arithmetic transfers. Unlike the remaining accumulators, it can also be used as the operated location for absolute-value arithmetic transfers. In a multiply operation, it serves as the high-order product, while in division it starts as high-order dividend and ends as the dividend remainder. It serves as a working register for coupled shift operations.

The digit register is composed of SR3 cards having three inputs and three outputs. The Bus 2 output is not used. The entire register can parallel transfer to and from the arithmetic bus registers in one digit time. The arithmetic bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The register can be shifted either right or left with the output at the low order or high order, respectively. The low-order serial output feeds the output latches, the high-order position for serial regeneration, the high-order positions of accumulator 2 and the word buffer register for shift operations. Separate capacitors are used in the low-order position with dump provisions to discharge the capacitors, when only the output latches are to be set. The high-order serial output feeds the low-order position of the word buffer register and floating point logic. Separate capacitors are used to permit dumping when the output is not being used.

The high-order position has an SR4 extender to provide additional serial inputs. The basic serial input is used as an entry from the adder output lines. The Input 3

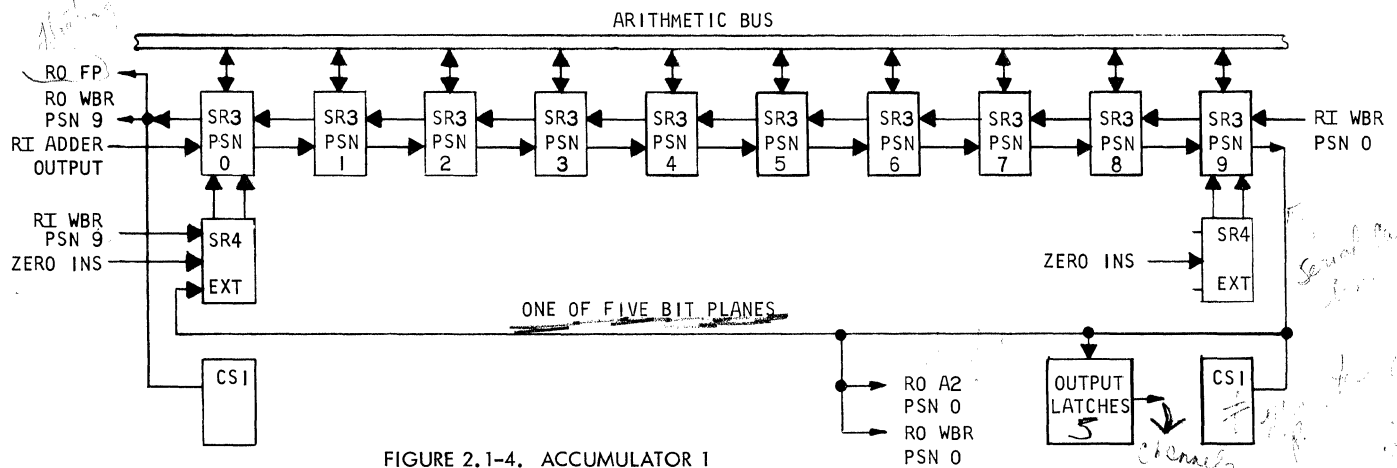


FIGURE 2.1-4. ACCUMULATOR 1

circuits of the basic register are used for serial left shift transfer. The extender inputs are used to serially enter from the low-order position of the word buffer register, to serially enter the register low order output for regeneration, and for zero insert during a shift operation.

The low-order position has an SR4 extender on the 1-bit and 2-bit lines to provide serial entry of zeros during left shift. The Input 3 entry of the basic register is used to serially enter the high-order position of the word buffer register.

SR1 cards are used for the three sign-position bits of the register. The serial read-out is used for regeneration only. Serial operation of the digit register does not operate the sign position. The sign position has separate read-in and read-out drivers. It is parallel-transferred to and from the arithmetic bus with the digit register or may be operated independently for sign adjustment during shift codes.

2.1.02 ACCUMULATOR 2 (Figure 2.1-5)

Accumulator 2 is a magnetic-core shift register with ten digit positions and a sign position. The register can be used as either an operated or addressed location in normal arithmetic transfers. In a multiply operation it serves as the multiplier register giving way to the low-order product. In division it starts as the low-order dividend and ends with the quotient.

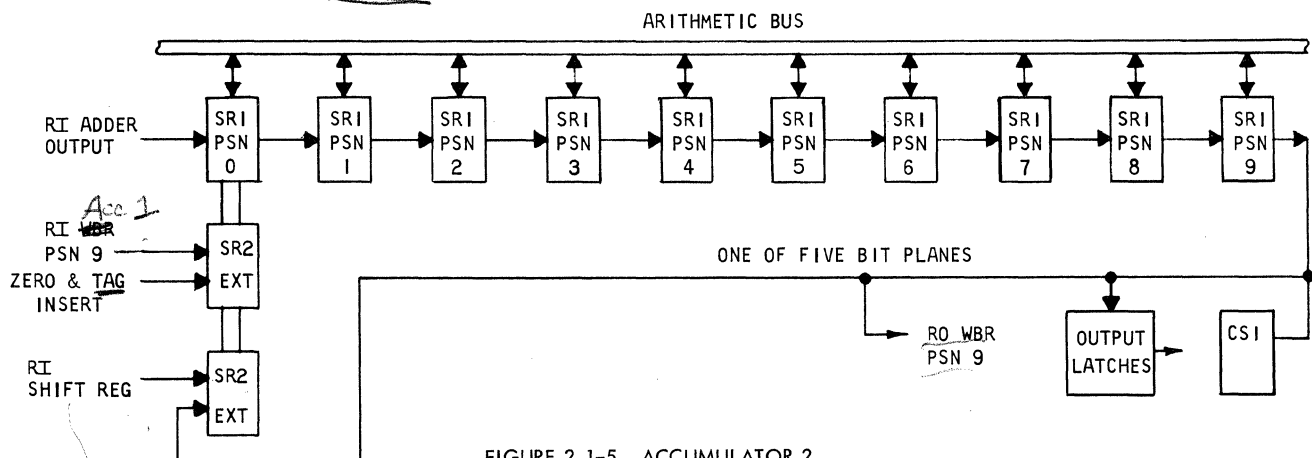


FIGURE 2.1-5. ACCUMULATOR 2

The digit register is composed of SR1 cards with two inputs and outputs. The entire register can parallel transfer to and from the arithmetic bus registers in one-digit time. The arithmetic bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The register can be read out at the low-order position with a right serial shift. The serial output feeds the output latches, as well as, the high-order position for serial regeneration, and the low-order position of the word buffer register for shift operations.

The high-order position has two SR2 extenders to provide four additional serial inputs. The basic serial input is used for input from the adder output lines. One extender is used to serially enter the register low-order output for regeneration and to insert quotient digits from the shift register. The second extender is used to insert zeros and five-bit tags, as required, and to serially enter the low-order output of the word buffer register.

SR1 cards are used for the three sign-position bits of the register. The serial read-out is used for regeneration only. Serial operation of the digit register does not operate the sign position. The sign position has separate read-in and read-out drivers. It is parallel-transferred to and from the arithmetic bus with the digit register or may be operated independent for sign adjustment during shift operations.

2.1.03 ACCUMULATOR 3 (Figure 2.1-6)

Accumulator 3 is a magnetic-core shift register with ten digit positions and a sign position. The register can be used as either an operated or addressed location in normal arithmetic transfers. In a multiply operation it serves as the multiplicand, while in division it serves as the divisor. Accumulator 3 serves the special function of Ramac Address Register for all disk storage operations.

The digit register is composed of SR1 cards with two inputs and two outputs. The entire register can parallel-transfer to and from the arithmetic bus registers in one-digit time. The arithmetic bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The register can be read out with a right serial shift at the low-order position. The serial output feeds the output latches and, in addition, the register high-order position for serial regeneration. Separate capacitors are used in the low-order position with dump provisions to discharge the capacitors when only the output latches are to be set.

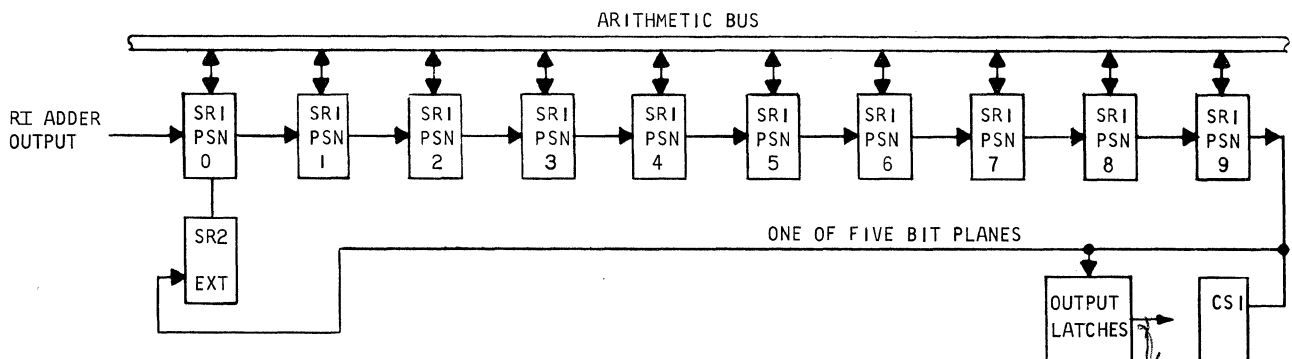


FIGURE 2.1-6. ACCUMULATOR 3

Channel 1 or 2

The high-order position has an SR2 extender to provide an additional serial input. The basic serial input is used as an entry from the adder output lines. One input is used in the extender to enter the register low-order serial output for regeneration.

SR1 cards are used for the three sign-position bits of the register. The serial read-out control is used for regeneration only. Serial operation of the digit register does not operate the sign position. The sign parallel-transfers to and from the arithmetic bus simultaneous with the digit register transfer. The same read-in and read-out drivers are used for both sections.

2.1.04 AUXILIARY REGISTER (Figure 2.1-7)

The auxiliary register is a magnetic-core shift register with ten digit positions and a sign position. Functionally this register is considered an instruction processing element, but it also serves in most arithmetic operations. The register layout differs from other ten position registers in that it is broken into three sections labeled Auxiliary I, II, and III. These sections have two, four, and four positions, respectively. Arithmetic operations use the full ten positions as a single register. In block transfers to or from core storage, Auxiliary II and III function as two 4-position registers for start and stop addresses. In index processing, Auxiliary II, or II and III, may function as four- and eight-position registers, respectively.

The digit register is composed of SR3 cards having three inputs and three outputs. The entire register can parallel transfer to and from the information bus and core storage during one memory clock cycle. The information bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The entire register, or sections II (and/or III), can be right serial shifted to provide outputs from positions five and nine. The serial output from position 9 feeds output latches and, in addition, serial regeneration at positions 0, 2, and 6. The serial output from position five feeds a second set of output latches and, in addition, serial regeneration at position 2. Separate capacitors are used in both of these read-out positions to allow dumping the capacitor charge when no serial regeneration circuits are being used.

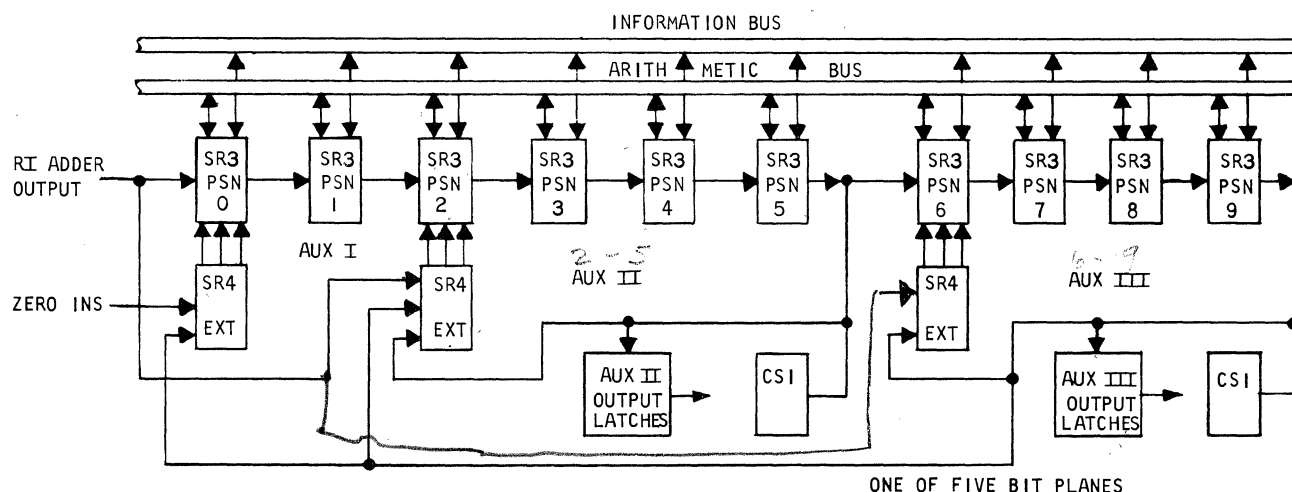
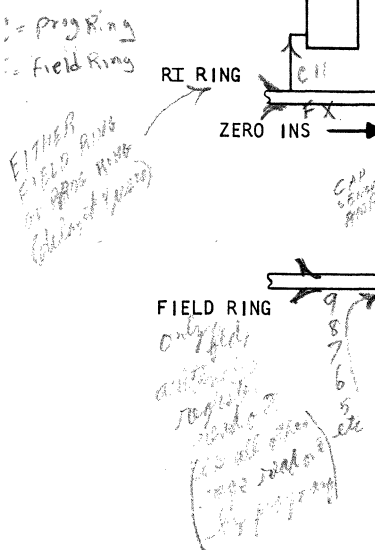


FIGURE 2.1-7. AUXILIARY REGISTER

SR3 cards are used for the three sign-position bits of the register. The serial read-out is used for regeneration only. Serial operation of the digit register does not operate the sign position. The sign position has separate read-out and regeneration drivers. The sign is parallel-transferred to and from the information and arithmetic busses with the digit register. It may also read-out independently to the arithmetic bus for sign analysis. Sign insertion is provided with bit insert drivers, which feed the serial capacitors.

The arithmetic register is a magnetic-core shift register with ten digit positions and a sign position. The register serves as the input and output buffer between core storage and the arithmetic area for data words. All applications of field control are made in the arithmetic register. All block transfers under control of the arithmetic and program unit pass through the register. This includes unit records to and from input/output and console typewriter.



10

The digit register is composed of SR3 cards with three inputs and three outputs. The entire register can be parallel transferred to or from either the information bus in one memory clock time or the arithmetic bus registers in one digit time. The respective bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register except when a serial or parallel entry is required.

The arithmetic register does not have a true serial shift. Input and output in right serial shift form are effected by logic switching by progressively scanning the individual positions with a digit ring. Each position has separate read-in and read-out control drivers to permit blanking and inserting during read-in from the adder output lines. For read-out, gated capacitor sense amplifiers are connected to each serial read-out capacitor. These capacitors are charged each digit time for parallel regeneration. The coincidence of a ring-digit-pulse and sensed bits sets the output latches. Advance of the digit ring causes the next position to the left to be read on the following digit and so on.

Each register position has a zero insert latch associated with it. At the start of a serial operation all of the latches are set. As a digit is scanned into a register position, its zero insert latch is reset. At the end of the serial operation, except add to memory or store, each position, in which the latch is still set, receives a parallel zero insert. The insert is effected by blanking the serial read-out and inserting bits into the serial capacitors with bit insert drivers.

SR3 cards are used for the three sign-position bits of the register. The serial read-out is used for regeneration only. The serial scan circuits of the digit register do not operate the sign position. The sign position has separate regeneration and bit insert drivers. The sign is parallel-transferred to and from the information and arithmetic busses with the digit register. The sign can read out to the arithmetic bus independently for arithmetic sign analysis. The sign can be blanked to insert an arithmetic sign for the computed result.

2.1.06 WORD BUFFER REGISTER (Figure 2.1-9)

The Word Buffer Register is a magnetic-core shift register with ten digit positions and a sign position. It serves as the buffer for unit record and typewriter input-output, the data register in multiply and divide operations, and as the working

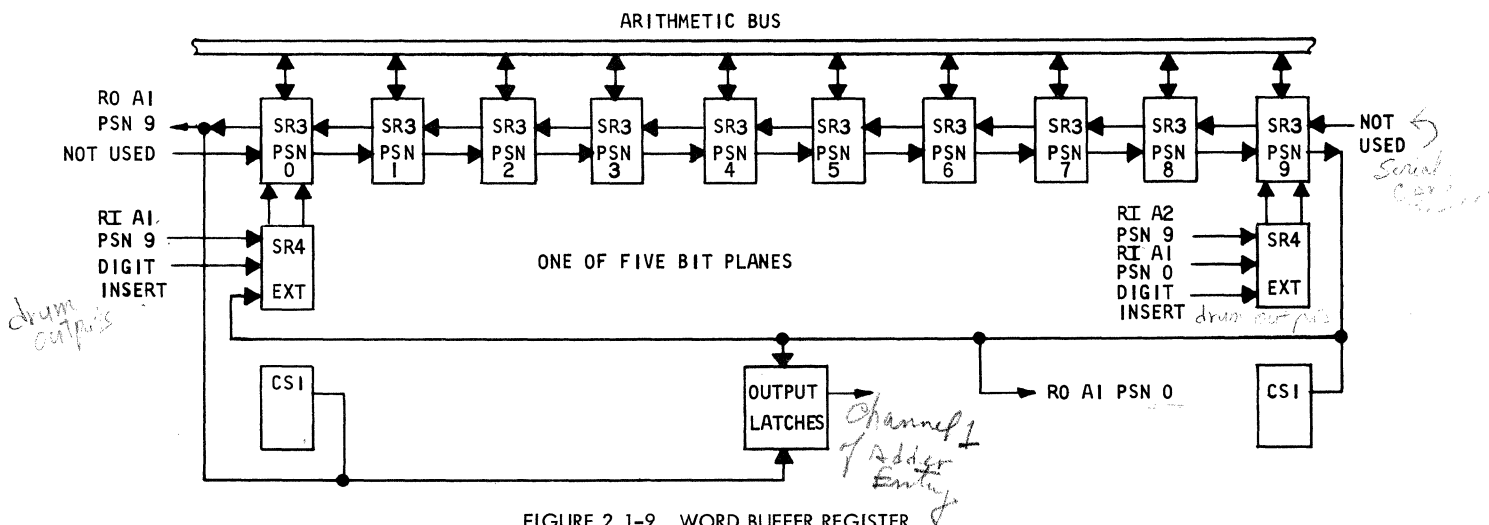


FIGURE 2.1-9. WORD BUFFER REGISTER

register in coupled shift operations. Its use in arithmetic operations arises from its ability to shift both left and right, and its insert provisions.

The digit register is composed of SR3 cards having three inputs and three outputs. The Bus 2 output is not used. The entire register can parallel transfer to and from the registers on the arithmetic bus in one-digit time. The arithmetic bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift is required. The register can be shifted either right or left with the output at the low-order or high-order, respectively. The serial outputs from both ends feed the same set of output latches. Separate capacitors are used to allow dumping the charge if only the output latches are to be set. The high-order output also feeds the low-order of Accumulator 1 for shift operations. The low-order position feeds the high-order position of the register for serial regeneration on right shift, and the high-order position of Accumulator 1 for shift operations.

The high-order position has an SR4 extender to provide additional serial inputs. The basic serial input is not used. The input 3 circuits of the basic register are used for serial left shift transfer. The extender inputs are used to serially enter the register low-order position for serial regeneration, the low-order position of Accumulator 1 for shift operations, a digit insert which includes Channel 2, Manual Inquiry from the drum, and the arithmetic register output latches for numeric to alphabetic conversion.

The low-order position has an SR4 extender to provide additional serial inputs. The basic input 3 is not used. The extender inputs are used to serially enter the low-order position of Accumulator 2 and the high-order position of Accumulator 1 for shift operations. The third extender input is used for digit insert which includes the drum and console typewriter entries, and the arithmetic register output latches for alphabetic to numeric conversions.

SR3 cards are used for the three sign position bits of the register. The serial read-out is used for regeneration only. Serial operation of the digit register does not operate the sign position. The sign position has a separate regeneration driver and bit insert drivers. The sign is parallel-transferred to and from the arithmetic bus with the digit register. It can be blanked to insert the sign during the serial entry from the input/output area.

Information entered into the word buffer register from a keyboard enters at a keyboard controlled rate -- not the 4 usec clock rate.

2.1.07 ADDRESS START REGISTER (Figure 2.1-10)

The address start register is a magnetic-core shift register with four digit positions. It serves as a transfer register between positions two to five of the arithmetic bus and the computer address bus during block transfers. The register is also used to read in the address location from console typewriter.

The register is composed of SR3 cards with three inputs and three outputs. The entire register can be parallel-transferred to ^{CAB}~~or from either the arithmetic bus~~ (positions 2-5) or the computer ^{arithmetic} address bus in ~~one digit time~~. The respective bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The register can be left serial shifted for serial read-in at the low order or

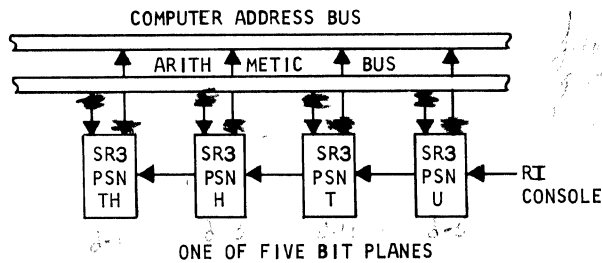


FIGURE 2.1-10. ADDRESS START REGISTER

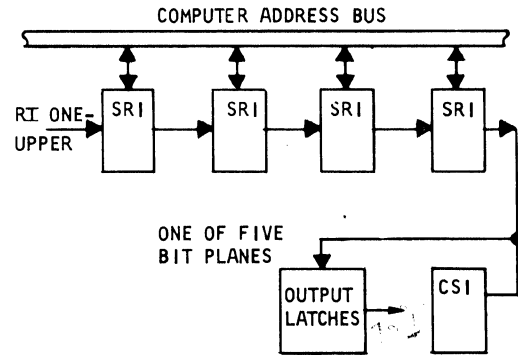


FIGURE 2.1-11. ADDRESS CONTROL REGISTER A

units position. No serial output is provided. Because the register is blanked prior to the serial read-in and only four serial digits can be entered, no provision has been made for dumping the thousands position serial capacitor.

No special inputs or extenders are used on this register. The digit count and control during the serial transfer are a function of the console.

2.1.08 ADDRESS CONTROL REGISTER A (Figure 2.1-11)

The address control register A is a magnetic-core shift register with four digit positions. It serves as the address register for the record definition word during block transfers under arithmetic and program control.

The register is composed of SR1 cards having two inputs and two outputs. The entire register can be parallel-transferred to and from the computer address bus in one digit time. The computer address bus capacitors are used for the transfer. The serial read-out is used to parallel regenerate the register each digit time except when a serial shift or parallel entry is required. The register can be read out at the low-order position with a right serial shift. The serial output feeds through capacitor sense amplifiers to output latches located in the One-Upper input circuit. ~~Separate capacitors are used on the units position to allow dumping the capacitor charge after sensing.~~

No special inputs are used on this register. The serial input on the thousands position reads in the output of the One-Upper. A five-bit tag digit is read in from the One-Upper at the start of each serial transfer. When the tag is read out of the units position, the operation is stopped and the new address is ready to parallel-transfer the location of the next record definition word to the computer address bus.

2.2.00 TRANSMISSION LINES

Data is transmitted from one area to another or from register to register by a group of lines designated as a bus or channel. In some cases, these lines connect only between two devices and serve no external function, and thus, are not named. In general, the transmissions are made either in Core Mode from the Shift Registers or in CTDL Mode. Validity check circuits are applied to most of the major transfer lines to insure usable data.

The core mode lines include the parallel transfer capacitor used by the register system. On long lines the shield adds to this capacity and the basic capacitor value is

adjusted to prevent overload of the read-out circuits. During a transfer, the line becomes an intermediate storage. The information bit is stored in the combined capacity of the line. During the transfer, the charge can be sampled by use of an Emitter Follower Driver (DE), or a Translator (XL), to feed CTDL switches and latches, or for validity checks. The charge remaining on the capacitor is still ample to set a register core at any point along the line. If no core is set, an equivalent loading in the form of a dump control is used to discharge the line before the next transmission.

The CTDL transmission lines are normally fed by the static output of a latch. Logic can be fed directly during a transfer to gate or sample as required. Core mode devices can be driven by using bit insert drivers. Validity check circuits are used on most of the lines to insure valid digit combinations.

2.2.01 ARITHMETIC BUS (AB)

The Arithmetic bus is a core-mode transfer line used to distribute data within the A & P area. The bus parallel transfers ten-digit words with sign over fifty-three bit lines. Transfer on the arithmetic bus may occur on each four microsecond A & P digit time. The lines are charged at A-pulse time of the clock cycle and discharged at C-pulse time. The charge is on the lines for approximately two microseconds.

The arithmetic bus connects the following registers for a bi-directional transfer:

1. Accumulator 1	Full-word transfer
2. Accumulator 2	Full-word transfer
3. Accumulator 3	Full-word transfer
4. Auxiliary Register	<u>Auxr 2</u> or <u>Full word</u>
5. Arithmetic Register	<u>Full-word transfer</u>
6. Word Buffer Register	Full-word transfer
7. Address Start Register	RI Positions 2-5 only

The arithmetic bus is capacitor-sense sampled, during transfer, for reading into the Significant Digit Scanner and the Factor Sign Latches.

Transfers on the arithmetic bus are not validity-checked. The data is checked before use either in computation or transfer to core storage. The bus capacitors are provided with a parallel clear control, which operates each digit position that has no register read-in (RI SDS or Athr. Factor Sign only). The clear control also operates on non-transfer cycles to prevent a charge build-up on the capacitors.

Charge on all cycles or no Xfer cycles

2.2.02 INFORMATION BUS (IB)

The information bus is a core-mode transfer line used to distribute data and instructions between the A & P registers, core storage, and the tape-channel control units. The bus parallel transfers ten-digit words with sign over fifty-three bit lines. Transfer on the information bus may occur on each six-microsecond period of the memory clock. The lines are charged at U-pulse of the clock cycle and discharged at Y-pulse. Transfers in and out of the A & P area require a line up of the two clocks. The normal A-pulse register read-out must occur at U-pulse time and the C-pulse read-in at Y-pulse time. In all cases the bus capacitors are charged for approximately four microseconds.

ABOE

In the A & P area, the information bus connects the following registers for a bi-directional transfer:

- | | |
|------------------------|---|
| 1. Arithmetic Register | Full-word transfer |
| 2. Auxiliary Register | Full-word, Auxr 1 and 2, or Auxr. 1 and 3 |
| 3. Program Register | Full-word or RO 6-9 only |
| 4. Instruction Counter | Positions 2-5 only |

Positions sign and 0-5 are capacitor-sense sampled to set the logic latch program registers. These positions are also sampled for pre-sense of core storage access requirements. The full word is sampled to set the condition latches and the priority-mask latches. The setting of these latches can also be read to the information bus with bit-insert drivers. A zero insert is provided to fill in positions 0-1 and 6-9 when the instruction counter is read out alone for storage or display.

The information bus is validity-checked during each transfer. The validity check latch is gated by the various information gate signals. The bus capacitors have a clear or dump control to allow discharge when no core register is being read in. All transfers to core storage require a line dump. Transfers to the A & P program area have a selective dump control, because the word is split between core and latch registers as follows:

- | | |
|------------------------|-------------------------------|
| 1. Positions Sign, 0-1 | Instruction and IC entry |
| 2. Positions 2-5 | Instruction and TLU |
| 3. Positions 6-9 | IC entry and on an error dump |

2.2.03 COMPUTER ADDRESS BUS (CAB)

The computer address bus is a core-mode transfer line used to transfer addresses within the A & P area and to core storage. The bus parallel transfers a four-digit address over twenty-bit lines. Transfers on the computer address bus may occur on each four microsecond A & P digit time. The lines are charged at A-pulse and discharged at C-pulse of the A & P clock. The bus capacitors are charged for approximately six microseconds when data enters A & P, and two microseconds when data is being stored.

The computer address bus connects the following registers in the A & P area:

- | | |
|-----------------------------------|----------------------|
| 1. Program Register "D" | Four-digit transfer |
| 2. Instruction Counter | Four-digit transfer |
| 3. Address Start Register | Four-digit transfer |
| 4. Address Control A | Four-digit transfer |
| 5. Program Register Index (Latch) | Low-order two digits |
| 6. Program Register Field (Latch) | Low-order two digits |

A two-position zero insert provides the high-order digits for the transfer from the two-digit registers to satisfy bus validity. The bus is capacitor-sense sampled to set logic latches, known as CAB latches, which provide a static output. These latches are also used to detect a 999X address.

The bus has no validity check on transfers within the A & P area. Transfers to core storage are checked in the core-control area. The bus capacitors are provided with a parallel clear or dump control to discharge the lines on transfers where a core

register is not read in. The dump also functions on each non-transfer cycle to insure that the capacitors remain discharged.

2.2.04 ADDER OUTPUT CHANNEL

The adder output channel is a core-mode transfer line used to transfer the results from the adder to the various data and processing units within A & P. The channel serial transfers digit by digit in two-out-of-five bit code on five-bit lines. The lines are charged at A-pulse time by bit insert drivers from the logic output latches of the adder. The individual registers discharge the lines at C-pulse time. The lines are charged for approximately two microseconds.

The adder output feeds the following registers (The major Op Codes for each are also listed):

1. Accumulator 1	Mult., Divide, Shift Codes
2. Accumulator 2	Multiply and Shift Codes
3. Accumulator 3	Mult., Shift, TLU Codes
4. Auxiliary 1	Multiply Code
5. Auxiliary 2	TLU, Transfer, Shift Codes, Index Codes, Unit Record Codes
6. Arithmetic Register	Add and Store Codes
7. Field Length Register	Field Control
8. Shift Register	Shift Codes
9. Program Register "D"	Index Cycle

The adder output is validity-checked during each digit time without gating. When the adder is not operated, the channel is fed with zeros during each digit time to maintain validity. These zeros are available for insertion in any of the connected registers. The capacitors on the channel are discharged by a clear or dump control, if the data is not read into a core register.

2.2.05 CHANNEL 1 (Adder Entry A)

Channel 1 is a CTDL serial transfer line that feeds one of the factors to the adder. The channel transfers digit by digit over five-bit lines. Data remains on the lines as long as the logic entries are gated. In operation, the logic latches feeding the channel remain fixed for a minimum of two microseconds during C-pulse time.

Channel 1 is fed by the following register units (The major Op Codes for each are also given):

1. Accumulator 3	Mult. and Disk File Codes
2. Auxiliary 2	Index Codes
3. Auxiliary 3	Add, Mult., and Transfer Codes
4. Arithmetic Register	Store, Mult., Add Codes
5. Program Register Data	Shift, Index Codes
6. Program Register Field (TP)	Field Control
7. Word Buffer Register	Mult., Shift Codes

Channel 1 also feeds the high order of the word buffer register for ~~Multiply~~ ^{Divide} and the disk storage controls for addressing. In addition to the register outputs, channel 1 is fed by logic to insert special digits for adjustments and entries.

Example

Channel 1 is validity-checked each digit time without gating controls. When the adder gating is off, zeros are inserted on the channel by logic to satisfy the validity test. No clear or dump control is necessary on a CTDL bit line since the line has no storage effects.

2.2.06 CHANNEL 2 (Adder Entry B)

Channel 2 is a CTDL serial transfer line used to feed one of the factors to the adder. The channel transfers digit by digit over five bit lines. Data remains on the lines as long as the logic entries are gated. In operation the logic latches feeding the channel remain fixed for a minimum of two microseconds during C-pulse time.

Channel 2 is fed by the following register units (the major Op Codes for each are also given):

1. Accumulator 1	Mult. and Divide Codes
2. Accumulator 2	Mult. and Divide Codes
3. Accumulator 3	Mult. and TLU Codes
4. Auxiliary 2	TLU, Transfer, and Index Codes
5. Auxiliary 3	Store, Mult., and Index Codes
6. Arithmetic Register	Add Codes
7. Program Register "D"	Shift, Index Codes
8. Program Register Field (UP)	Field Control
9. Shift Register	Shift Codes

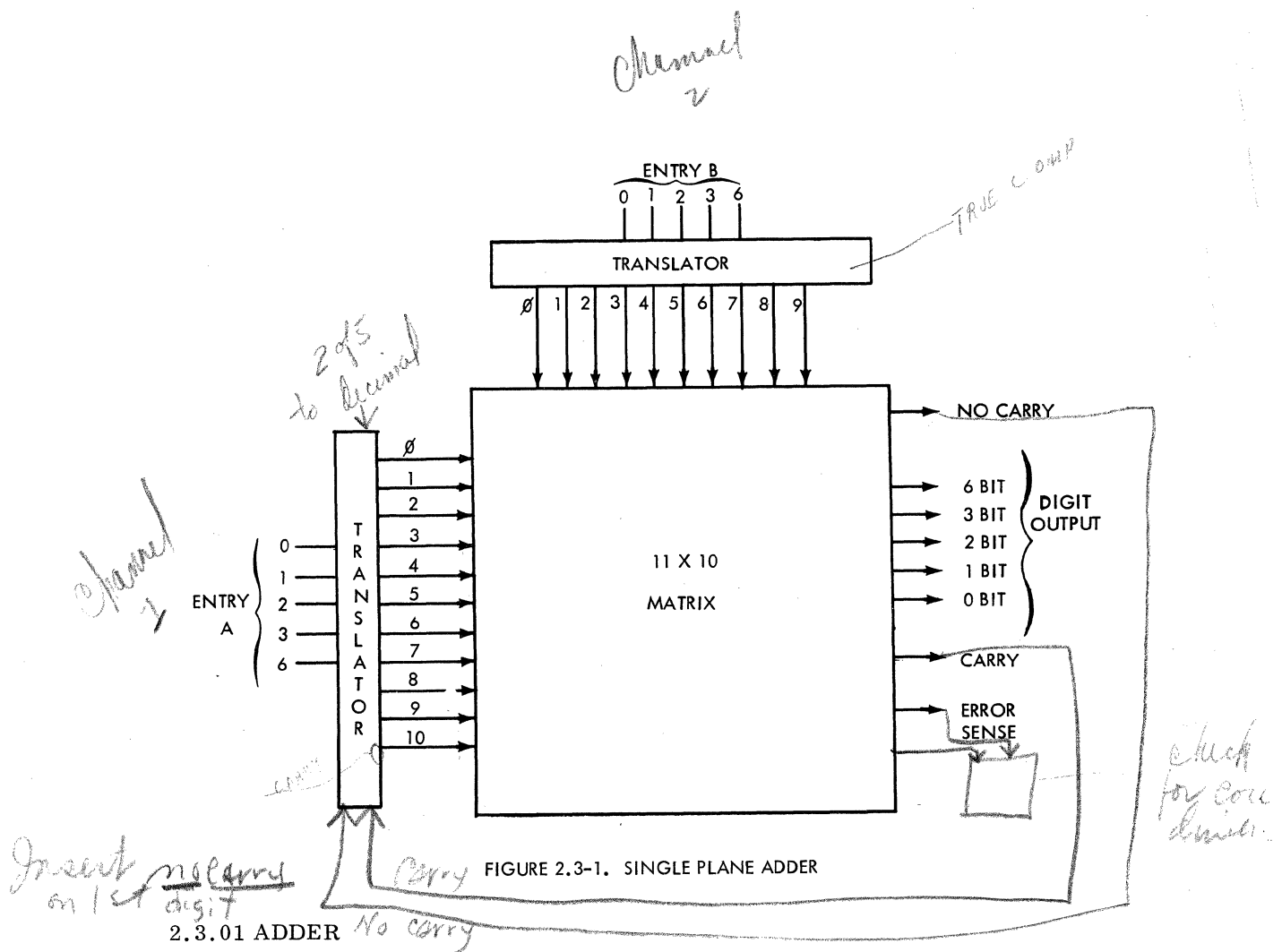
Channel 2 is also used to enter the number of the free access arm from disk storage. Special digits are inserted as required for adjustments and entries.

Example
Channel 2 is validity-checked each digit time without gating controls. When the adder gating is off, zeros are inserted on the channel by logic to satisfy the validity test. No clear or dump control is necessary on a CTDL bit line because the line has no storage effects.

2.3.00 PROCESSING UNITS

In addition to the basic data handling registers, the IBM 7070 has several units which are associated with the processing of data. There is little similarity between the individual units either in construction or operation. The units use magnetic-core components along with CTDL switches and latches.

In case of the adder a special core matrix is used with driver and sensing configuration similar to a small core memory device. The adder input and output circuits are CTDL. Two devices, the Significant Digit Scanner and the Shift Register, make use of shift register cards for special applications. The validity check and match devices use the validity check card (VCR), which is a magnetic-core component capable of detecting the number of logic bits. A final group of units, the field length register, the field ring, and the condition latches, are entirely CTDL and fall into the category of digit storage and control ring devices.



The adder, as its name indicates, is capable of combining two inputs to produce an arithmetic sum at the output. It operates serially one digit at a time, while combining the carry condition of the previous digit operation. The adder consists of an 11 X 10 single plane core matrix, using ribbon wound cores, and CTDL input and output circuits (Figure 2.3-1).

The inputs originate from channels 1 and 2 in the two-out-of-five notation. Each entry has a decimal translator to feed the core drivers. Adder entry A incorporates the carry and no-carry signals as gates. The translation is direct with the no-carry signal and plus one with the carry signal. The plus one results in an eleven line output from the entry A translator. Adder entry B is gated with the true and complement add signals. The true add signal provides direct translation to decimal, while complement add gives a nine's complement inversion. One of the pair of signals must be present at each entry for operation. Each input is sampled at the driver input with an adder sample pulse to control adder timing.

The adder matrix enters the eleven inputs from entry A on one axis and the ten inputs from entry B on a 90° axis (Figure 2.3-2). Cores are placed at the intersection of each pair of horizontal and vertical drive lines. This results in diagonal rows of cores representing a particular sum digit output. The sums represented by the intersections above the center diagonal are less than ten and thus produce no carry output. Below the diagonal line the sums are greater than ten and produce a carry. The matrix also contains two error cores, which check each axis for multiple drive lines. These cores read out to the error-sense circuit. A reset-bias line also feeds each core of the matrix with current in opposition to the drive lines.

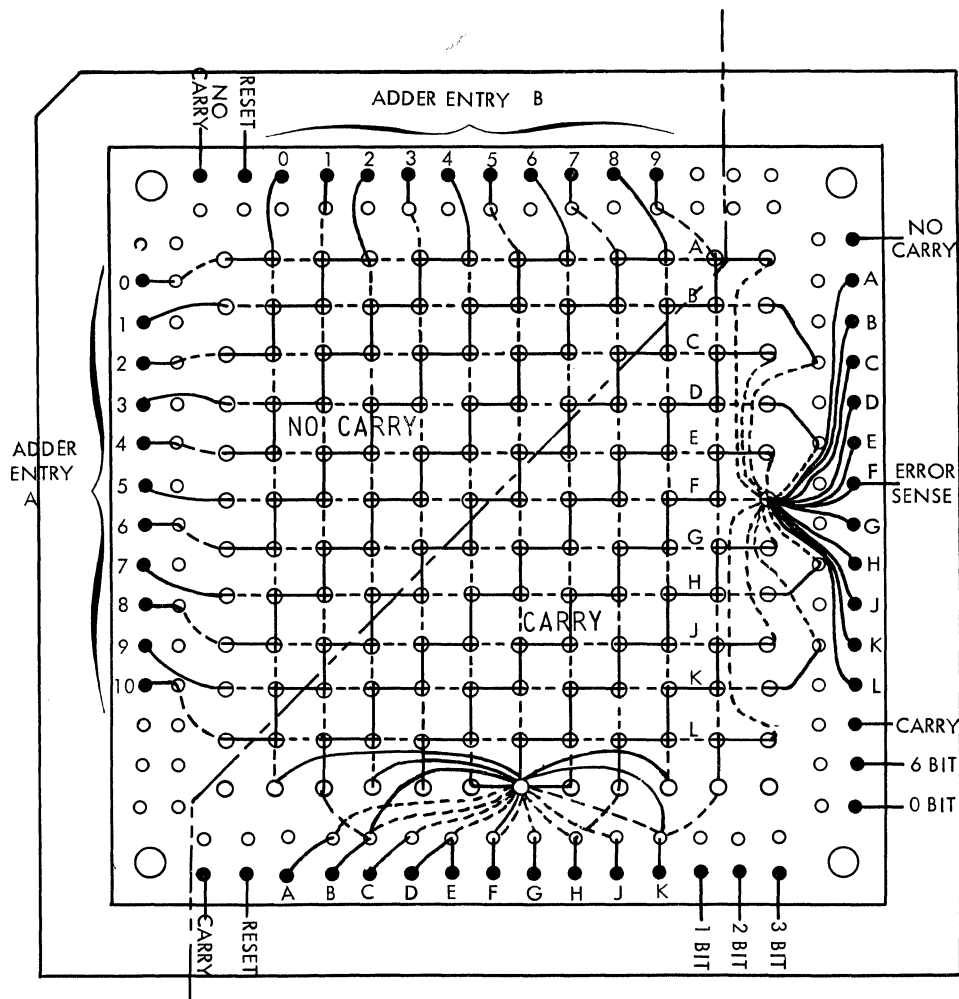


FIGURE 2.3-2. READ-IN WINDINGS--ENTRY A AND ENTRY B

The switching of an individual core in the matrix is the result of full switching currents on the two intersecting drive lines. Figure 2.3-3 shows a hysteresis loop with the relative effects of the input currents. The reset-bias current moves the effective current from point A to point B. Each core with a single-drive current returns the level back to point A. A second-drive current on the core moves the current level to point C to set the core. When the drive currents are removed, the current level returns to point B and the core is reset.

Each core is wound with three sense windings. Two are used to feed the digit output in two-out-of-five notation; the third winding is either carry or no-carry depending on the core location. Each sense line feeds a sense amplifier, which is bi-polar to accept pulses of either polarity. A pulse is available on the sense line as the core sets, and a second pulse as it resets. With a combination of these signals, the sense amplifier output remains on for 1.5 to 1.8 microseconds. The output is used to set the output-first latches to provide approximately C to C gates. A second latch called output-second is set at AP time for an A to A gate. Figure 2.3-4 shows the approximate timing relations.

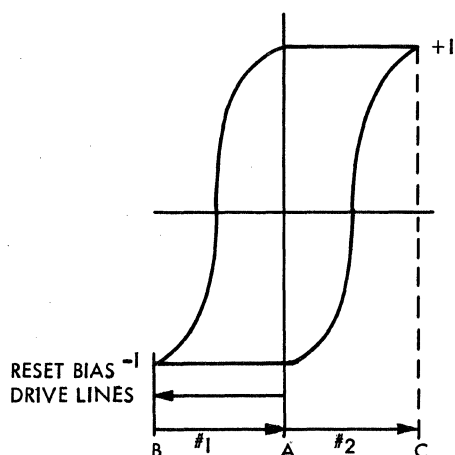


FIGURE 2.3-3. HYSTERESIS LOOP

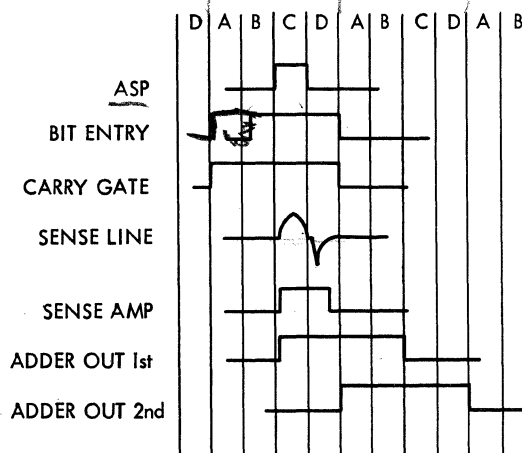


FIGURE 2.3-4. ADDER TIMING

The carry and no-carry first latches are used for detection of operation conditions such as compare, complement adjust, and divide overdraw. The carry and no-carry second latches are used to gate the next digit input to adder entry A. The adder-output-first latches are used to feed the insert drivers that feed the adder output channel. The adder-output-second latches feed a CTDL output and the validity check. A logic zero and non-zero output provide a means of detecting equals in compare operations.

In operation the factor digit from channel 1 and the carry-second line logic switch to effect a half adder. The carry condition is inserted for the first digit. For a true-add operation, a no-carry insert is normally used. With complement-add, a carry insert is normally used for ten's complement and a no-carry insert for nine's complement. Factors of any size can be added but are usually limited by the ten-digit word length.

Assume a simple true-add operation with a factor of 0675 on channel 1 and 0862 on channel 2. The 5 from channel 1 is switched with no-carry to produce a drive on the 5 line. This combines with the 2 from channel 2 to produce a digit output of 7 and no-carry. During the second digit time, the 7 from channel 1 switches with no-carry to remain a 7 and enters the matrix with the 6 from channel 2 to produce a digit output of 3 and a carry. For the third digit time the 6 from channel 1 switches with carry to drive the 7 line. With the 8 from channel 2 we produce a digit output of 5 and a carry. One more digit cycle is taken to enter the carry unless the accumulator area is filled. The 0 from channel 1 is switched with carry to drive the 1 line. With the zero from channel 2 we produce a digit output of 1 and no-carry. The sum as read out is 1537. The result digits return to the register one digit after the factor digit leaves.

A complement-add operation is identical except that the channel 2 factor would read into the adder as 9137. A carry insert is normally forced for the first digit. The sum would read out 9813 which is a complement value. This is determined by the no-carry output on the last cycle. The complement factor is corrected by entering it on channel 2 with zeros inserted on channel 1. Complement-add and carry insert are again used. The factor enters the adder as 0186 and after combining with the carry insert reads out as 0187.

The details of the channel 1 and channel 2 circuits, as well as the adder-output channel, are given in Section 2.2.00.

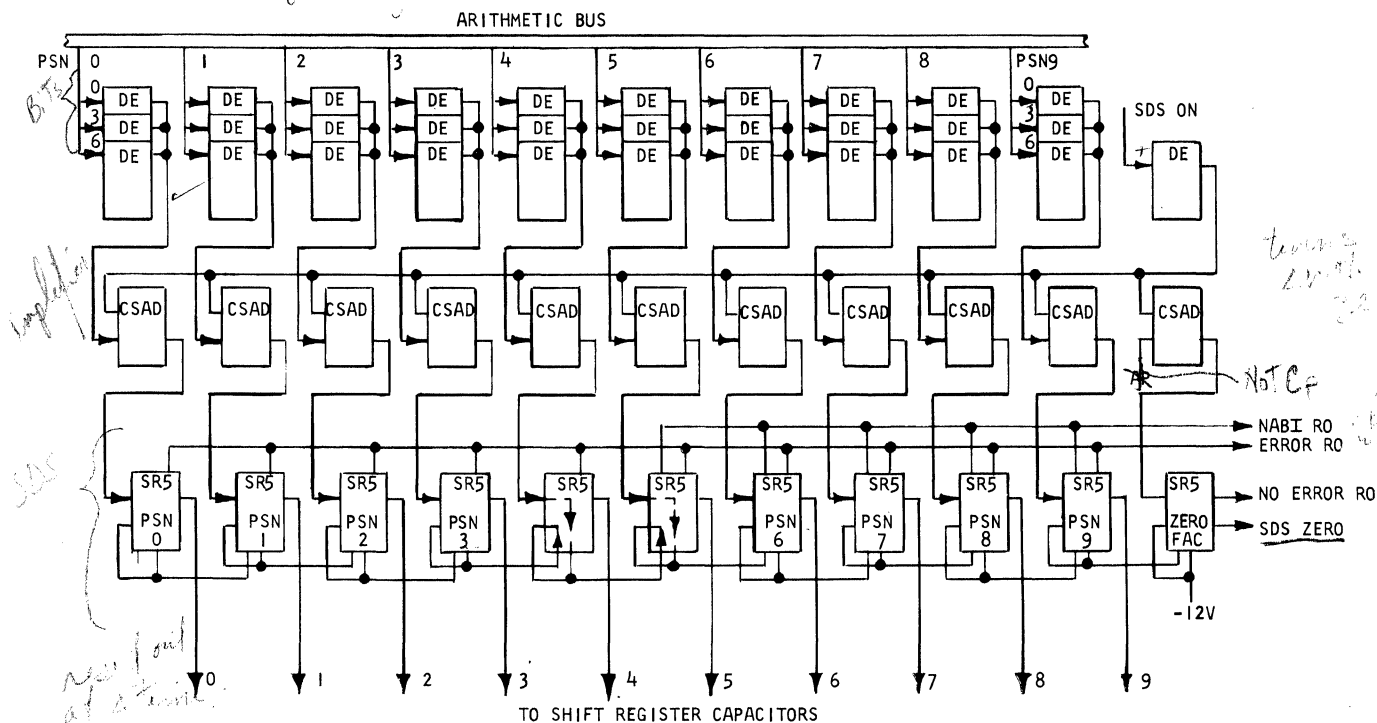


FIGURE 2.3-5. SIGNIFICANT DIGIT SCANNER

2.3.02 SIGNIFICANT-DIGIT SCANNER (Figure 2.3-5)

The significant digit scanner is a magnetic-core shift register used for processing. It differs from data registers by having only one bit plane. Eleven magnetic cores are used to represent the location of the high order significant digit position. Ten of the cores show the position within the word, while the eleventh represents all zeros. Only one core is set on a read-in.

On add-subtract operations the significant-digit scanner is used to determine the size of the factor in the operated accumulator. The output of the scanner is used in conjunction with the shift register and the field control system to control the length of the adding cycle. At the end of the adding cycle serial transfer, the auxiliary register contents are fed to the scanner. In this case the output should be from the zero core to indicate a correct add-subtract operation. Any other core output indicates a failure.

The significant-digit scanner is used for four other operations.

1. The operated-accumulator factor is scanned on shift and count operations, for the number of zeros to the left of the high-order significant digit. The scanner is used with the shift register to determine the value and to count the shift.
2. In a similar manner, the scanner is used for conversions between numeric and alphabetic. A special output is provided on the low-order five core positions for blank insert operations.
- 3,4. The factor is tested for zero in + 10, + 20, + 30, and + 44 codes.

NOTE: The 7070 Reference Manual (Form A24-7003) shows the mnemonics BZ# for operation codes plus 10, 20, and 30, and BXN for operation code plus 44. Actual machine logic (ALD) identifies these codes by the mnemonics TRZ# and TRNZI, respectively.

The significant-digit scanner is composed of eleven SR5 cards using a special input connection. One winding is driven from a logical OR with the presence of a 0, 3, or 6 bit on the bus position. This current is applied to the core winding in the set direction. The second windings on all cores are connected in series but are reversed to appear as reset or inhibit currents. Each of the driven windings is returned to the series connected chain at a point which feeds as inhibit current to all of the cores in the lower order positions. The inhibit current in the lower order positions prevents them from setting even if driven. With a significant digit in any position of the factor, the eleventh or zero factor core is inhibited. If no significant digit exists in the factor, the zero factor core is set from the sample pulse on the drive winding.

The significant-digit scanner is normally blank. When a bit is set, it is immediately read out with the read-out driver without regeneration. Depending on the operation, one of the three read-out controls is selected to make use of the information. In operations involving the shift register, the scanner resembles an extender. The shift register serial capacitor is used for the transfer. The zero factor core reads out to sensing circuits only. For the zero-non-zero test, the zero factor core shows the zero condition. A read-out from any other core is used to indicate an error on a Field End Test. On an ENB (+57) operation output, only the five low-order factor positions are connected together. The remaining cores are not used. All of the sense circuits use external capacitors for read-out.

2.3.03 SHIFT REGISTER (Figure 2.3-6)

The shift register is a magnetic-core shift register used for processing. Functionally it serves as a digit ring in several operations. It has eleven core positions arranged in a single plane. Shift or advance may be either to the left or right. The ten low-order positions are given the decimal values of 0 to 9. The eleventh or high order position is lettered X and serves to complete the transfer to the A first latch on a left shift. When it is used with the significant-digit scanner, the position numbers correspond to the digit positions of the word. The position numbering remains the same for operations involving digit count--the value being raised or lowered by the shift operation.

Output latches at either end signal the end of serial shift for control purposes. A set of parallel output latches read out the decimal value with a translation to two-out-of-five notation. These output latches read out in complement because of the nature of their use in major operations. References to single-core-shift-register (SCSR) and multiplier-quotient register (MQ) in the logics are the shift register and its parallel output latches, respectively.

The shift register is used in add-subtract operations, with the significant-digit scanner, to reduce adding time. The bit location is right-shifted as the factor is read out to the adder until the bit reads into the A-last output latch. The left shift and count operation makes similar use of the register but uses a left shift to count out the number of high order zeros. A specified shift sets the shift register from the computer address bus latches. The register is right shifted to count the specified shift value.

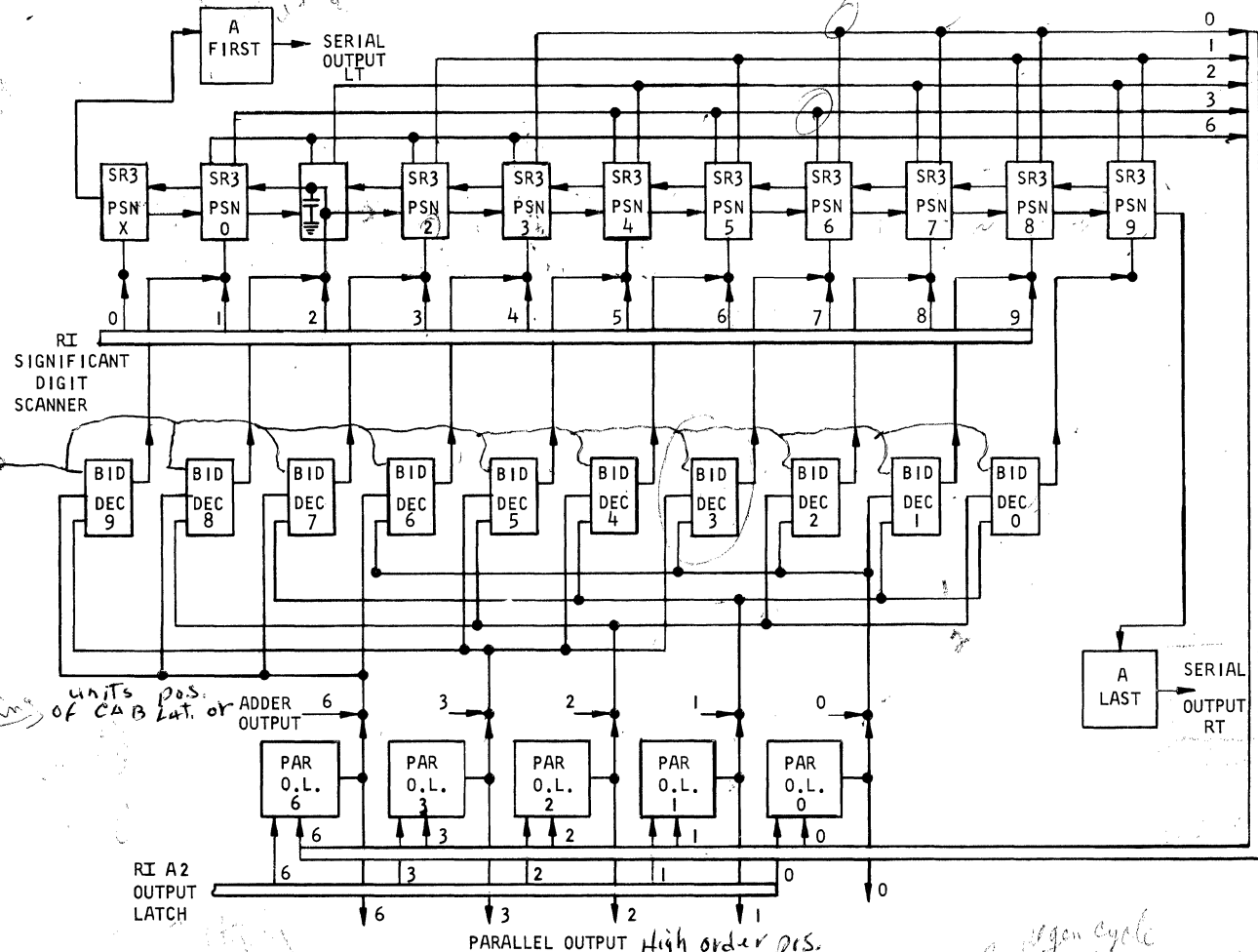


FIGURE 2.3-6. SHIFT REGISTER

In a multiply operation a multiplier digit is read into the parallel output latches from the accumulator 2 output latches. The digit value is analyzed for factors in the output latches. It is then read into the cores and right-shifted by the value of the factor multiple. At the end of the cycle it again reads into the output latches for factor analysis. The operation repeats until the bit reaches the A-last serial output latch.

In division the operation is similar, the output latches are set for zero at the start of each quotient digit. The latch value is read into the cores with a single left shift. When the bit is read back to the output latches, the value is raised by one. The process is repeated for each reduction cycle.

In numeric-to-alphabetic conversion operation, the shift register serves as a translator and static output of the significant-digit scanner read-out.

The shift register is composed of eleven SR3 cards. The serial entry is used for right shift control and the input 3 entry is used for left shift. The bus 1 input is not used. Entry to the register is always through the serial transfer capacitor. Read-

in may be under regen, right, or left shift control. Right shift is normally used. When the entry is from the significant-digit scanner, the serial capacitor of the position to the left is used. With a right shift, the bit enters the same digit position. Entries from the shift register parallel output latches or other two-out-of-five lines are translated and read in in complement. A right-shift entry causes the subsequent read-out to be one digit lower. A left shift causes an increase in the digit value. After read-in, the bit is regenerated each digit time that a shift or read-out is not made in the same manner as other registers. The output from the adder and the computer-address bus latches are OR'd with the parallel output latches for entry. In the multiply operation, the value in Accumulator 2 output latches is forced into the parallel output latches and then entered in the normal manner.

The parallel and bus 2 read-out windings are used together to provide the parallel output. The decimal value is translated to two-out-of-five notation by driving one of the two bit lines with each winding. This output is used only to set the parallel output latches. The latch value is again the complement of the decimal core position. The parallel output latches can be read out to accumulator 2 high order for quotient digit insert and to the adder entry for shift count storage. In the edit numerical to alphabetic with blank insert operation the latch output is switched with ring timing for control.

2.3.04 VALIDITY CHECK CIRCUIT (Figure 2.3-7)

A check for valid characters is made on many of the lines transmitting in the two-out-of-five bit code. The circuit used is a combination of a special magnetic-core component and logic. The output of the combination remains at the error level unless a valid condition is sensed. The circuit is used for both core-mode and CTDL inputs. On a bus or multiple-digit transfer line, the five lines for each digit are by separate circuits. The individual check circuits are combined in a common output load or switching network. A valid output indication is obtained only when all of the check circuits have a valid digit at the same time.

The circuit is composed of the magnetic-core component, which has both the cores and a switching load circuit (VCR). A special driver is used to set the core from the bit lines (CSAD). The core elements are current sensitive. Below an optimum current value no core is set, and the output remains at the static level. At an optimum current value one core is set and the output level goes negative momentarily. With larger current values a second core is also set and again the output remains at the static level. The output is controlled by diode switching. The first core circuit is biased positive and the second core is biased negative. The read-out of a set core opposes its bias condition. The input drivers which form the bit lines are collector OR'd. When coupled to the core circuit, the output currents are additive. Any two of the drivers turned on produces the optimum current to set only the first core.

Because the output pulse of the core circuit is of short duration, and at negative logic level, a latch is used on the output. The latch is set by a clock or gating pulse in advance with the static core output holding the latch-back. With a valid digit or word, the latch-back is broken and the latch is reset. An error is detected by switching the output of the latch after the check has been made.

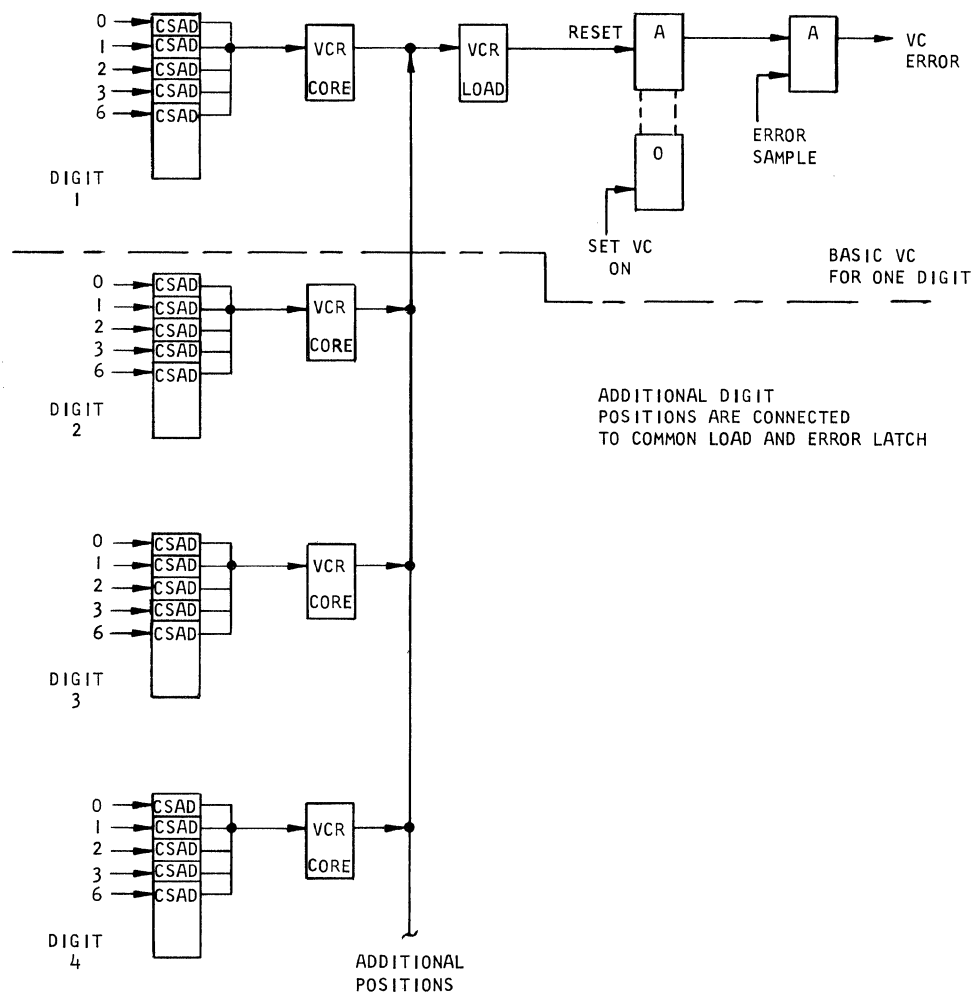


FIGURE 2.3-7. VALIDITY CHECK CIRCUIT

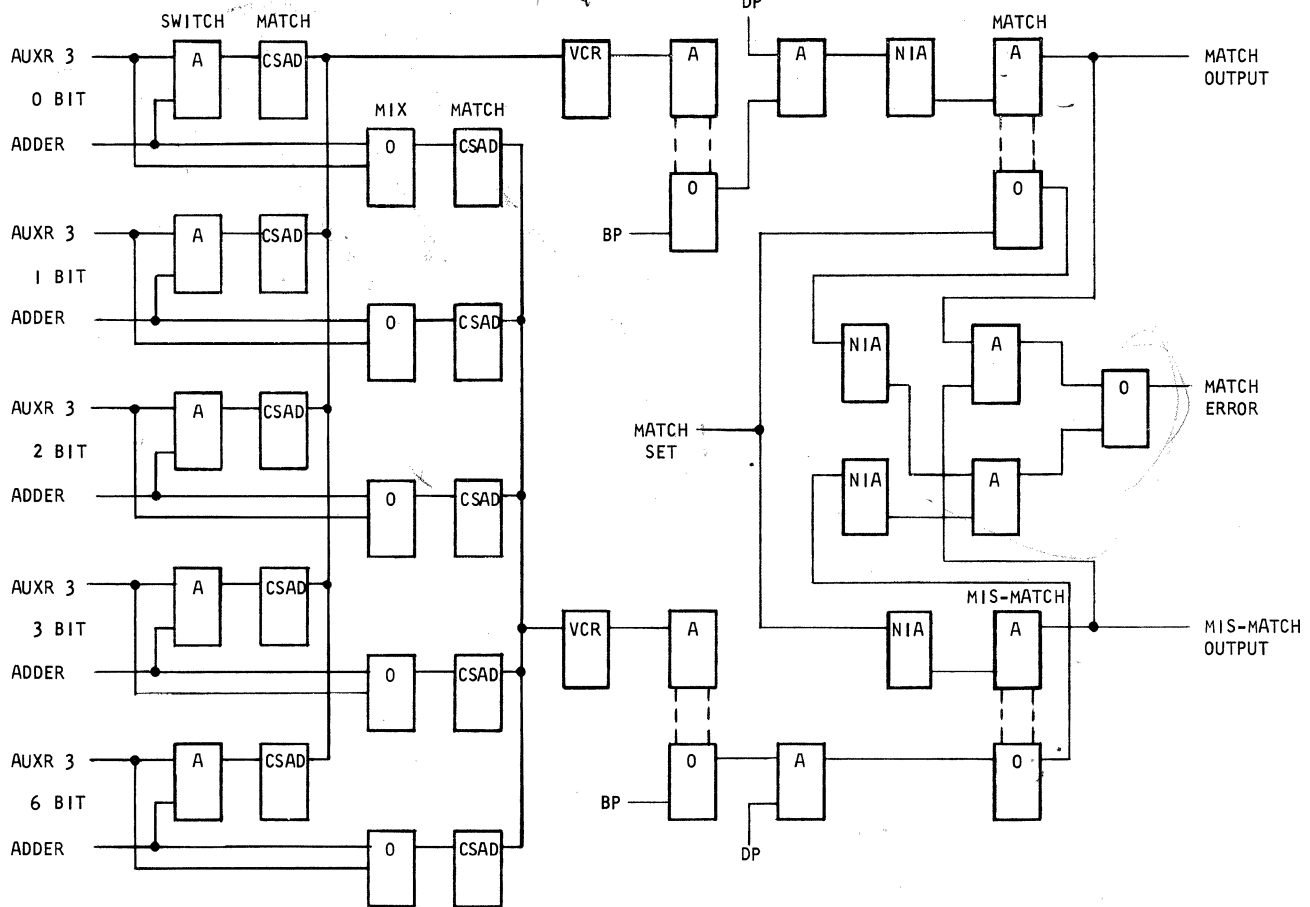


FIGURE 2.3-8. A&P MATCH CIRCUIT

2.3.05 MATCH CIRCUIT (Figure 2.3-8)

The match circuit used in the process channel control is a special application of the validity-check circuit. Each time the start address of the record definition word is up-dated, it must be checked for a match with the stop address. Two validity-check circuits are used in combination to check for a match, as well as a possible validity error. For one check the matching bit lines of the two inputs are switched to feed the five validity-check drivers (CSAD). The second check is made by mixing the matching bit lines to feed the five drivers of the second circuit. With a match condition both circuits indicate validity. On a mis-match both circuits indicate the equivalent of a non-valid condition. If the circuits operate independently, it shows a match error indicating an input validity error.

The sampled output of the two check latches is used to control the match and mis-match latches. Prior to a match check, the match latch is set and the mis-match latch is reset. With a match between the two inputs and no validity error, both of the check latches are reset. At sample time there is no signal to either break the latch-back of the match latch, or to set the mis-match latch. In a serial operation each digit is tested in sequence. If all positions match, the output of the latches remains as set. Failure to match in any position reverses the latches so that a mis-match is indicated. A mis-match in several positions does not further change the latches. If

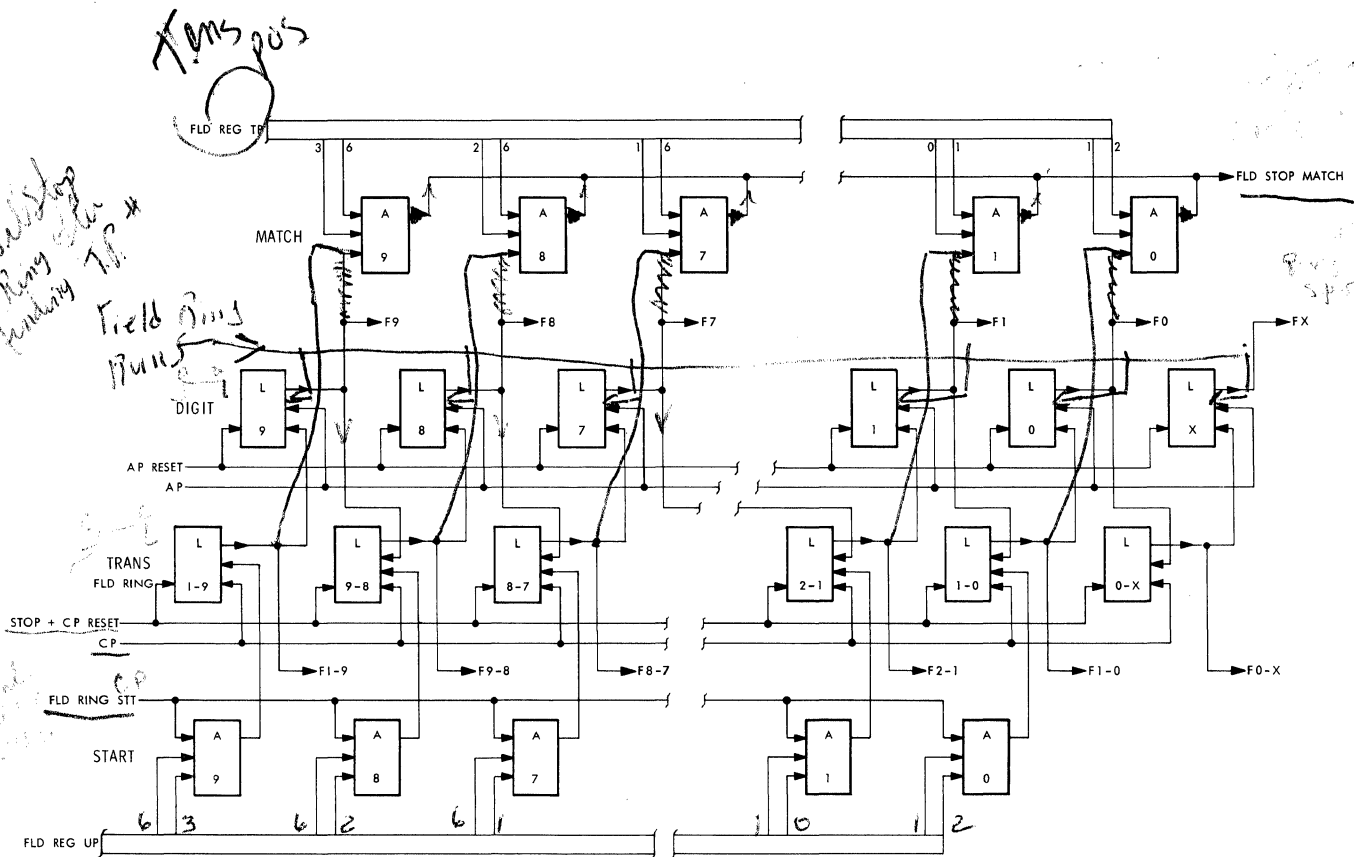


FIGURE 2.3-9. FIELD RING

the bit condition is in error with either missing or extra bits, only one output latch is reversed. The condition can only exist at the end of the operation if all other digits match. A check is made at the end of the operation with logic circuits. If both latches are set or both are reset, a match error signal is developed.

2.3.06 FIELD RING (Figure 2.3-9)

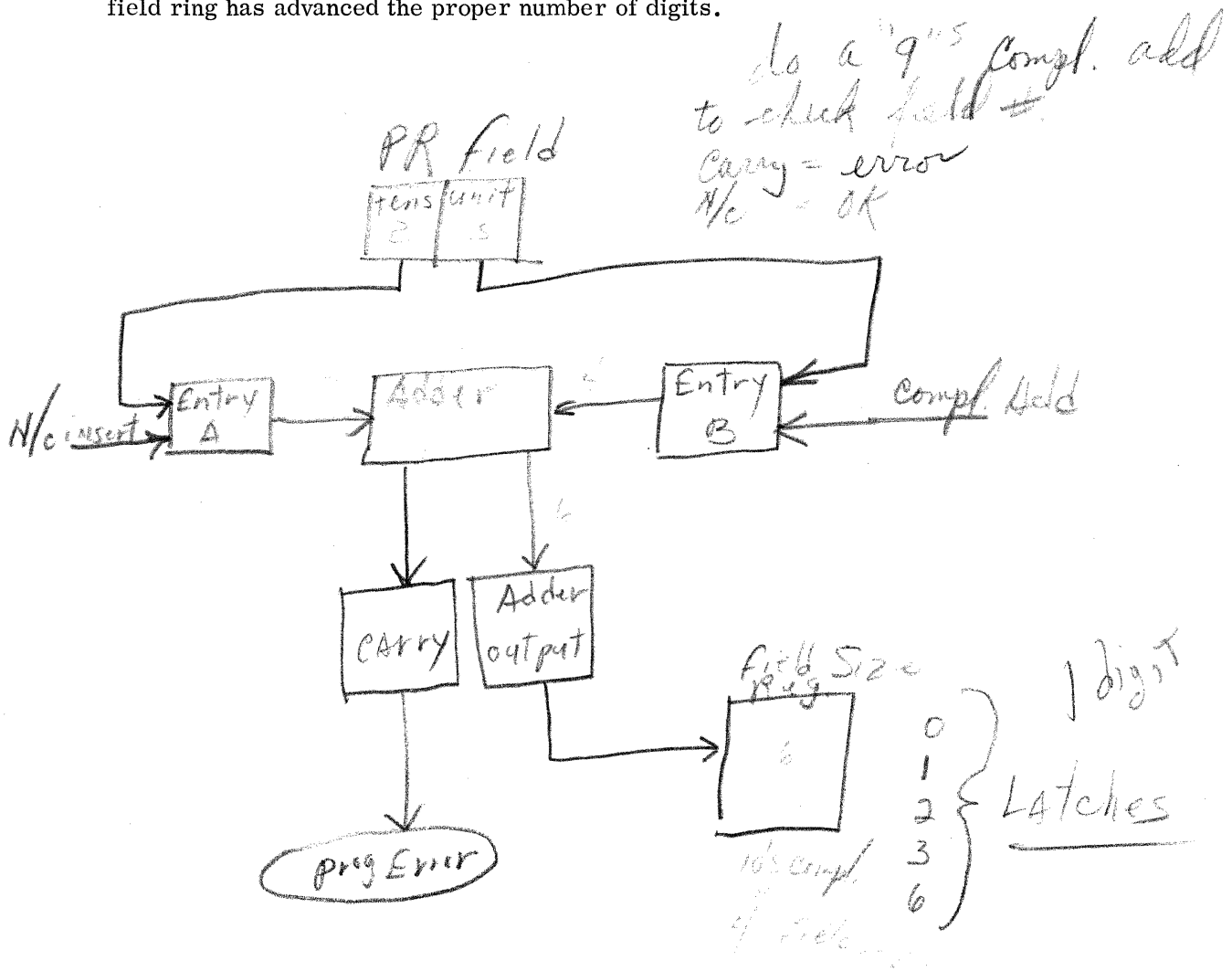
The field ring is basically a timing ring of the logic latch configuration. The ring is composed of eleven digit time latches and eleven transfer or half time latches. Each position remains set for 4 usec. The digit pulses are timed for A-to-A in successive digit times. The transfer latches are timed for C-to-C overlapping the digit times. The digit positions are numbered starting with F 9 and progressing down to F 0 and F X to correspond with the normal serial transfer of a word. The transfer pulses are numbered to show the digits between which they fall (F 6-5). The starting pulse is named F 1-9. The prime purpose of the field ring is to time the serial selection of digits from the arithmetic register.

The field ring can be started in any position. It advances digit by digit until stopped. The start and stop positions are under control of the program field register. The register is set either from the instruction word for field control or the value is forced into the register for special operations. The units position of the register indicates the starting point for the ring. The transfer, or C-to-C latch immediately before the desired position is set at C-pulse to permit control of the operation. At the following A-pulse, the digit, or A-to-A latch, sets. The transfer and digit latches continue to set in sequence at C-pulse and A-pulse, respectively. The ring continues until the digit position value matches the tens position of the field register. With a match the field ring advances one more digit position to allow replacement of the last adder output digit in the arithmetic register.

The field ring is also used in shift-from-point codes. In these operations the field register is set from the shift value in the computer address bus latches. The field ring functions in a manner similar to the shift register as a second serial shift count. The arithmetic register is not used for these operations.

2.3.07 FIELD LENGTH REGISTER

The field length register is a one-digit logic latch register. It contains five latches for storage of a decimal digit in two-out-of-five notation. The registers only use is in a check of the field ring operation on field control. The register is set from the adder output of the difference of the start and stop digits from the program field register. After the field ring has advanced to the match location, the field length register is checked against the program ring digit value. A match indicates that the field ring has advanced the proper number of digits.



field length 4 digits

3.0.00 INSTRUCTION PROCESSING ELEMENTS

3.1.00 CLOCKING

3.1.01 BASIC SYSTEM

Control of the operation timing in the A & P boxes is from the system primary timing or clock circuits. Clocks throughout the system are started simultaneously by a clock reset pulse. The clocks continue to advance in synchronism under control of the 500 kc sync pulses. These pulses are transmitted to the various areas with special drivers and balanced lines to obtain the minimum deviation between clocks. Figure 3.1-1 shows the general configuration when using the drum for clock control.

Each slide or pair of gates in the 7601 has a separate four microsecond basic clock to control the timing within the tower. Separate clocks are used to insure good timing relations between slides. The clock cycle is divided into four divisions designated as A, B, C, and D. Each division has the nominal timing of one microsecond. The rings operate continuously in a closed loop so that A follows D to form a four microsecond digit cycle. A digit cycle represents the time required to cycle in the core-register system. Data can be serially shifted one position or parallel-transferred between registers during each clock cycle. Several digit control rings, which are driven by the basic clocks, count groups of clock or digit cycles for specific operations. Controls that require accurate switching are further sampled by one of the basic clock pulses. The four A & P clocks are checked against each other during each D-time to insure synchronism.

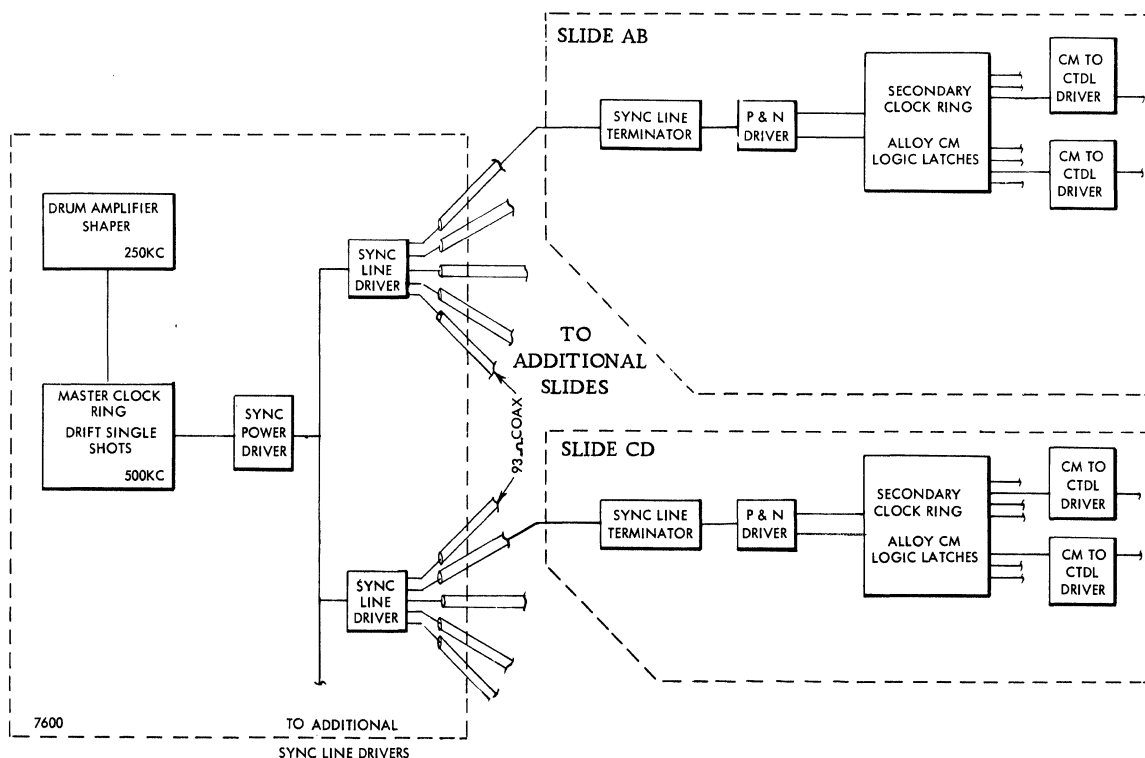


FIGURE 3.1-1. 7070 CLOCKING SYSTEM

The memory clock, which is also of prime importance to the A & P operation, is driven by the same 500 kc sync pulse. It is a six microsecond cycle with six one-microsecond divisions labeled U, V, W, X, Y, and Z. Because of the difference in the cycle time, it is necessary to check coincidence of pulses between the clocks for a memory access. A prescribed alignment of the clocks occurs every twelve microseconds. The alignment for a transfer from an A & P register to memory differs by six microseconds from the alignment for a reverse transfer. Control pulses, developed by sampling the clocks, produce a Memory-In Pulse (MIP) for the first type of transfer and a Memory-Out Pulse (MOP) for the second.

3.1.02 BASIC CIRCUITS AND TIMING

A & P operations require clock pulses of several different durations. The register system requires pulses on the order of 0.9 microseconds maximum. The CTDL latches require set pulses from 1.5 microseconds minimum. The maximum desirable pulse for sequential logic switching is about 2.0 microseconds. Reset of logic latches requires a nominal 1.0 microsecond. The basic clock produces two pulse timings for each position. In theory these values are approximately 0.6 and 1.6 microseconds. The delays of the logic blocks within the clock and the pulse stretching in the various driver blocks result in pulse lengths of approximately the desired values.

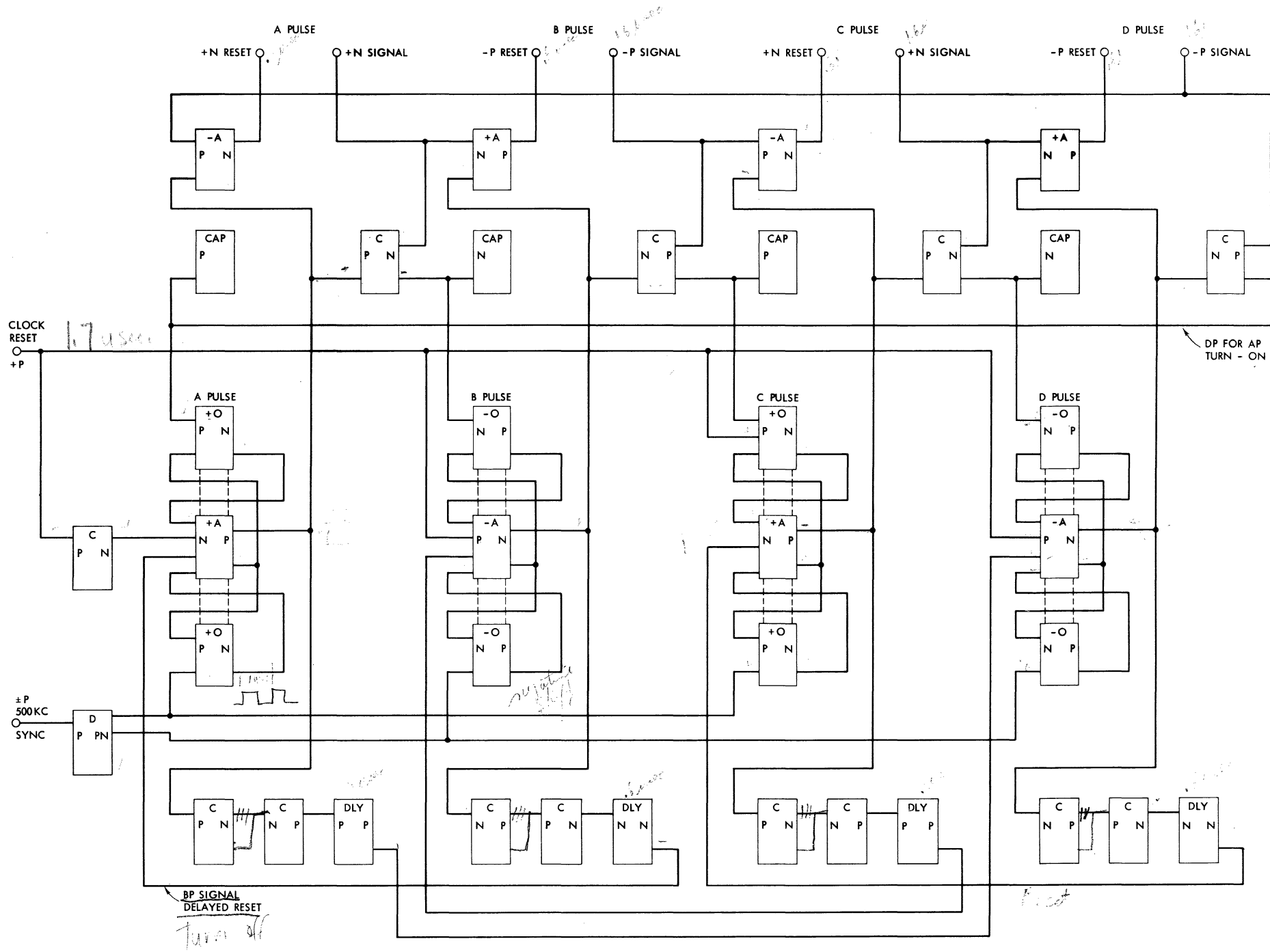
Figures 3.1-2 A and B show the general logic and timing for the basic four microsecond clock. Differences between individual clocks lay in the levels of the output signals and the powering for specific applications.

Modified current mode latches are used for each of the four pulse positions. The three-block configuration is actually a single latch with a switch input. A signal must be present at both of the OR blocks before the latch can be set. Latch-back is applied through both OR blocks to maintain the coincidence. This configuration provides the effective switching of three logic blocks with the delay of only two. The successive latches alternate positive and negative switching to make use of both the positive and negative excursions of the 500 kc sync pulse.

With the ring running, the A-pulse latch is set by the coincidence of the DP for AP turn on and the 500 kc sync pulse. This special DP consists of a separately buffered DP with a capacitor block added to delay the turn-off of the signal. The A-pulse latch is reset with an inverted B-pulse (Signal) which has been delayed 0.6 microseconds by a delay line block (BP Signal Delayed Reset). Thus, the A-pulse latch is on for a half-cycle of the sync pulse plus the 0.6 microsecond delay or a total of 1.6 microseconds. The resulting pulse becomes the A-pulse Signal output of the clock. It is used externally after powering as the A-pulse (AP) throughout the 7601. Within the clock the A-pulse signal is switched with the D-pulse signal to produce the A-pulse reset clock output. This pulse starts with the turn on of the A-pulse latch and ends with the turn-off of the D-pulse latch approximately 0.6 microseconds later. It is used for register drive pulses and reset lines for logic latches. The B-, C-, and D-pulses are developed in the same manner with the set and reset of comparable latches. A few signals are labeled special at the clock output. These are either signal or reset pulses with separate buffering or levels.

The clock reset control is used only in conjunction with computer reset. In a power-on sequence, the clocks do not have the logic requirements to start by themselves. In this case and when, through failure, a clock drops from synchronism, the reset must be used to start all clocks together. The clock reset pulse occurs during B and part of C (1.7 usec) of the primary clock. The A & P clocks are started at C-pulse by either

FIGURE 3.1-2A. A&P 4 USEC CLOCK



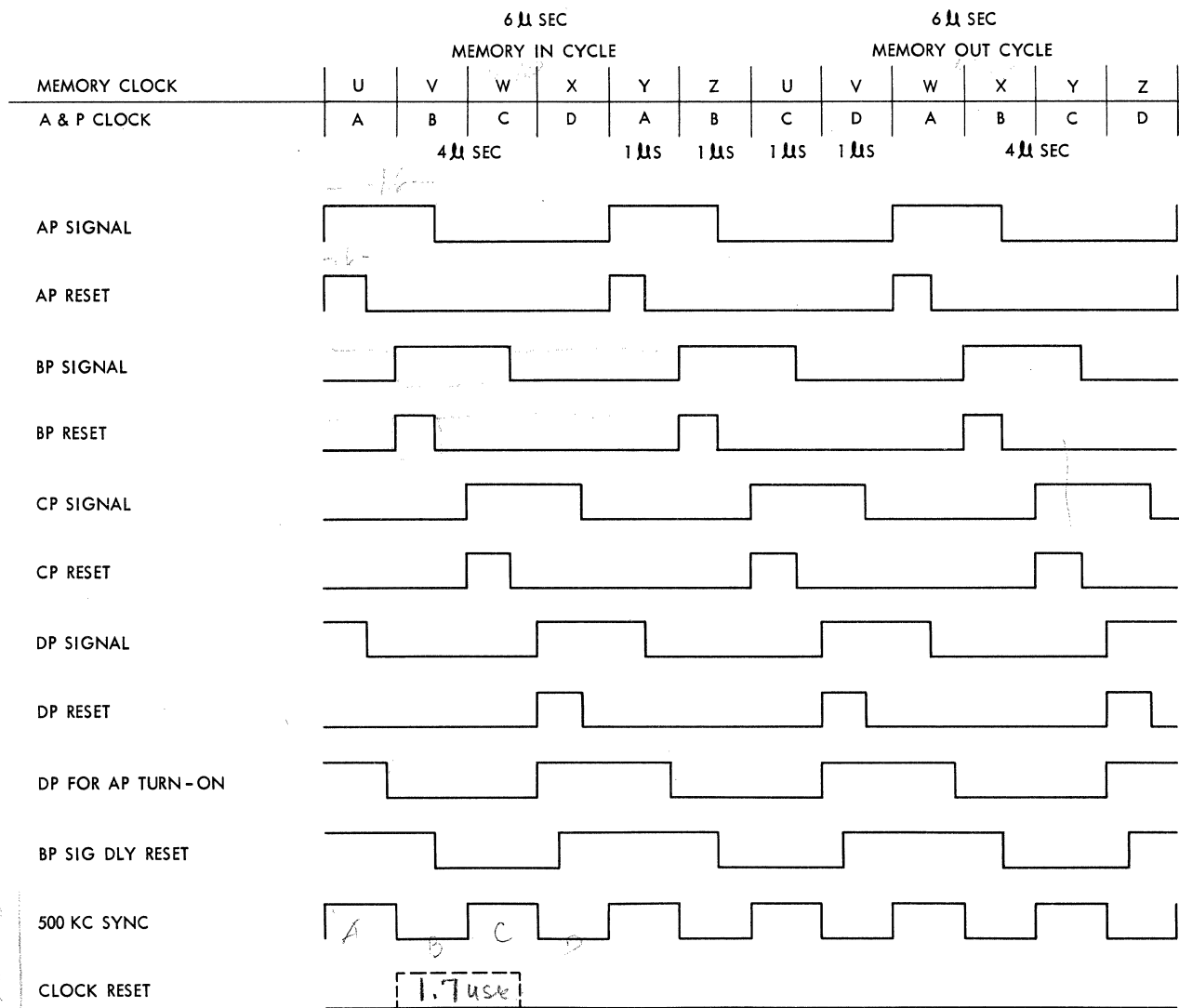


FIGURE 3.1-2B. A&P 4 USEC CLOCK TIMING

a reset or an inhibit of the A-, B-, and D-pulse latches with the clock reset pulse. The C-pulse latch is forced by providing an effective BP for CP turn-on signal. The inhibit is removed in time to set the D-pulse latch in the normal manner.

A group of early clock pulses are developed outside of the basic clock circuit. These are used to control some of the digit rings, which are used on more than one slide. The early A-pulse (EAP) is developed by delaying the D-pulse with a 0.4 microsecond delay line block. Thus, the pulse starts approximately 0.6 microseconds before A and ends approximately at the start of B. The early A-pulse reset (EAP Reset) is developed by delaying the D-pulse reset with a 0.4 microsecond delay line block. The pulse starts approximately 0.6 microseconds before A and ends at A. Other pulses of this type have similar development. The actual timing at the point of use may be somewhat later as the result of powering and transmission.

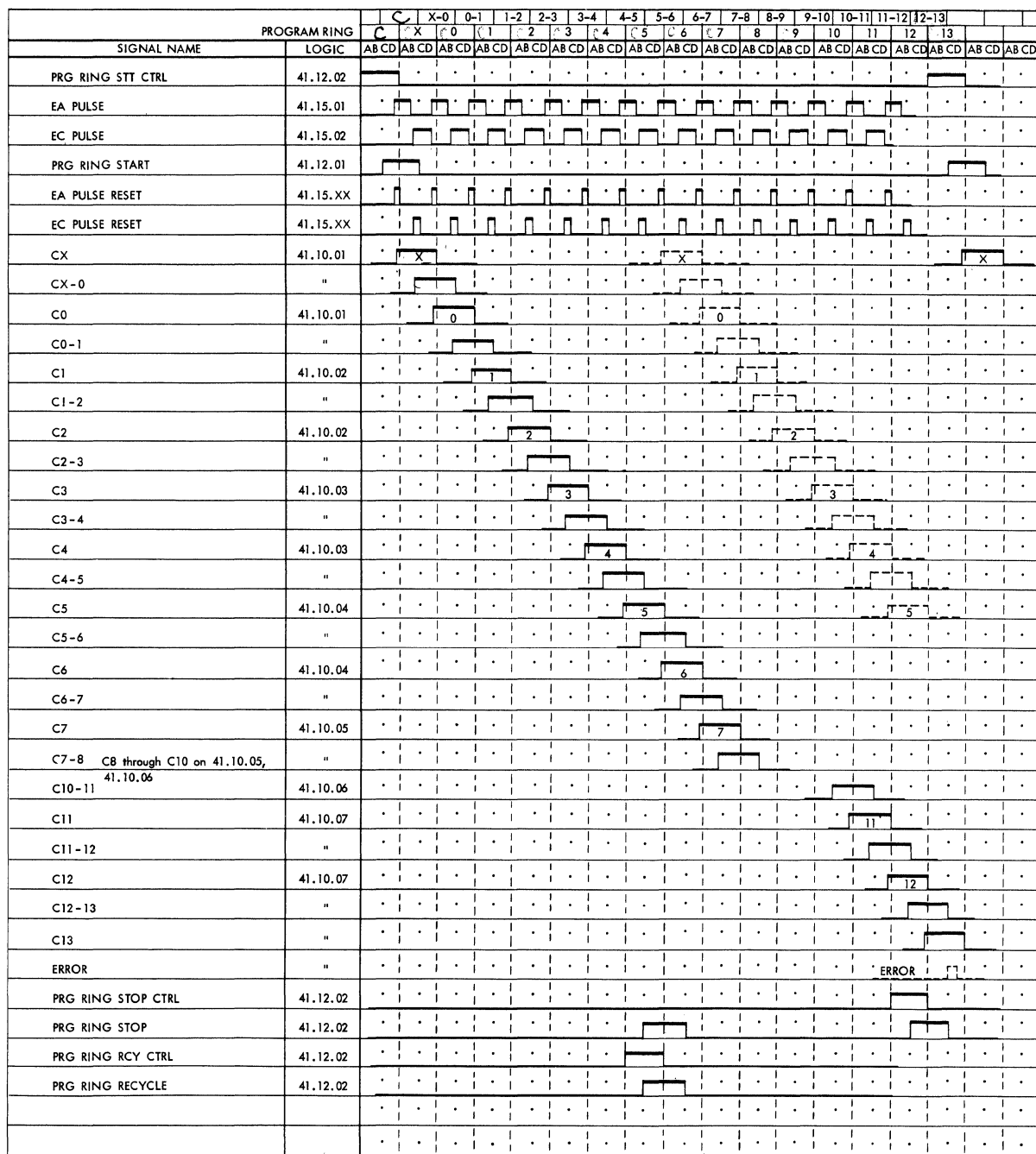


FIGURE 3.2-1. PROGRAM RING TIMING

3.2.00 CONTROL RINGS

3.2.01 PROGRAM AND PROGRAM CYCLE RINGS

The program ring is a digit timing ring of the logic latch configuration. The ring is composed of fifteen digit time latches and fourteen transfer or half-time latches. Each latch remains set for four microseconds. The timing of the various output pulses is shown in Figure 3.2-1. The digit pulses are timed for A to A in successive digit times. The transfer latches are timed for C to C overlapping the digit pulses. The digit positions are numbered starting with CX and C0 and progressing through C 13. The transfer pulses are numbered to show the digits between which they fall (C 1-2).

The program ring start serves as the first C to C latch for the start of the ring. The ring is always started at CX but may be stopped after any digit. The ring may be restarted or recycled at CX at any time. The program ring pulse serves as the C to C pulse and forces the ring start. The ring is stopped by setting the program ring stop latch at the desired digit time. The following transfer latch is not set and the ring stops at the end of that digit. Positive logic circuits for the program ring are shown in Figure 3.2-2. The objectives of the various control circuits are discussed with the appropriate operation code.

Working in conjunction with the program ring is a second ring known as the program cycle ring. Many operations require specific starting and stopping pulses for control. The program ring always starts at CX thus giving a specific starting pulse. The ring, however, may stop at any point making it difficult to select the proper stopping pulses. The program cycle ring fills this need for a pulse.

The program cycle ring is composed of logic latches. Three latches labeled Last, Test, and Stop + occur at digit time (A to A). Three other latches labeled Last -, Last +, and Stop lead the above respectively as transfer control (C to C). By starting the ring prior to the test time, the program ring is stopped by forcing the stop latch. The remaining positions of the program cycle ring then serve as stop pulses for the end of the operation. The early pulses of the program cycle ring serve as delays for those operations needing an extra digit time for adder carry processing.

3.2.02 CYCLE CONTROL RINGS (Figure 3.2-3, 4 and 4.1-2)

Either of the four control rings: instruction, index, data, or store, are started when an A & P address gate is returned to the A & P unit from core storage. The ring which is started depends upon whether an instruction request, index request, data request or store in memory initiated the memory request. Once started, the rings advance with A and C-pulses. The ring outputs are shown in Figures 3.2-3, 4. Both ring controls and ring outputs are discussed in later sections.

3.3.00 REGISTER UNITS

3.3.01 LATCH REGISTERS

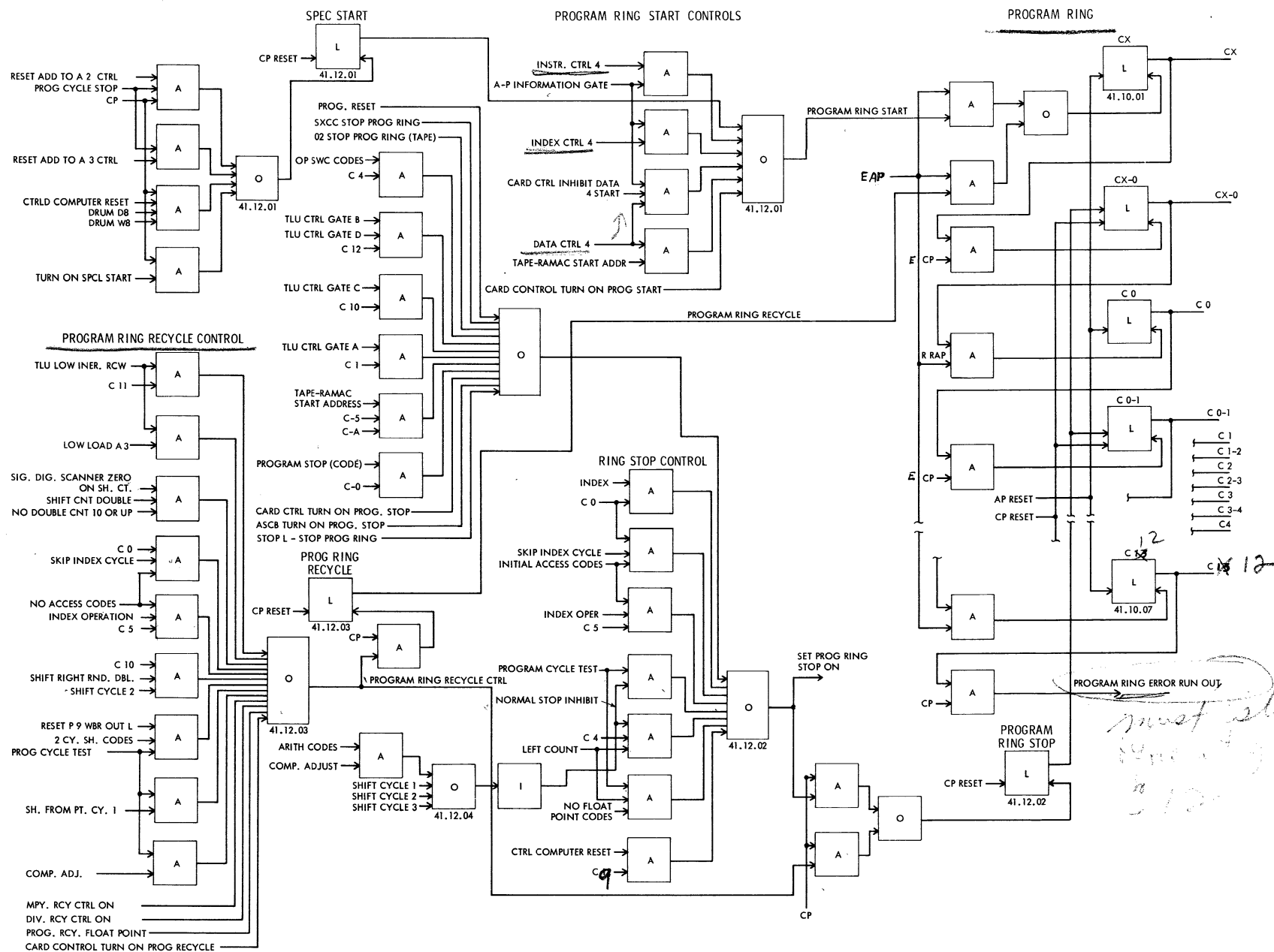
The operation, index, and field program register latches are turned on by sampling the information bus lines with a control gate. The development of the RI PR latch gates is shown in Figure 3.3-1. These are standard latches which provide a static output until they are reset.

The computer address bus latch register is a similar register which provides a static output of the D address for use in the operation matrix decode circuits. Figure 3.3-1 shows the controls for the CAB latch register.

3.3.02 CORE REGISTERS

The program register and instruction counter are standard core registers. Figures 3.3-2 through 5 show both instruction counter and program register, as well as the control circuits for both units. The gates used to control input and output are discussed under each operation which uses the program register and instruction counter.

FIGURE 3.2-2. PROGRAM RING AND CONTROLS



[illegible]

FIGURE 3.2-4. STORE CONTROL RING TIMING

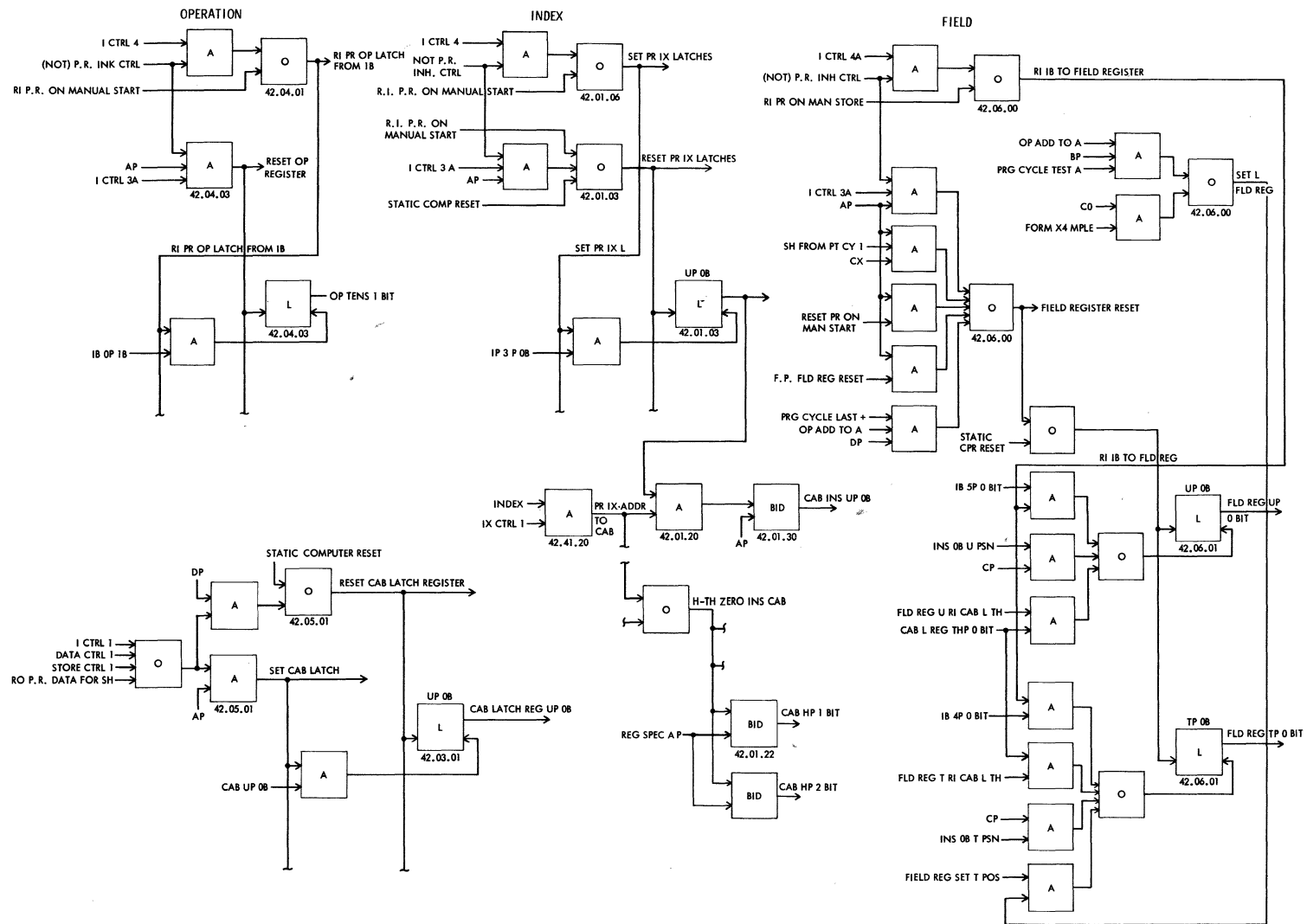
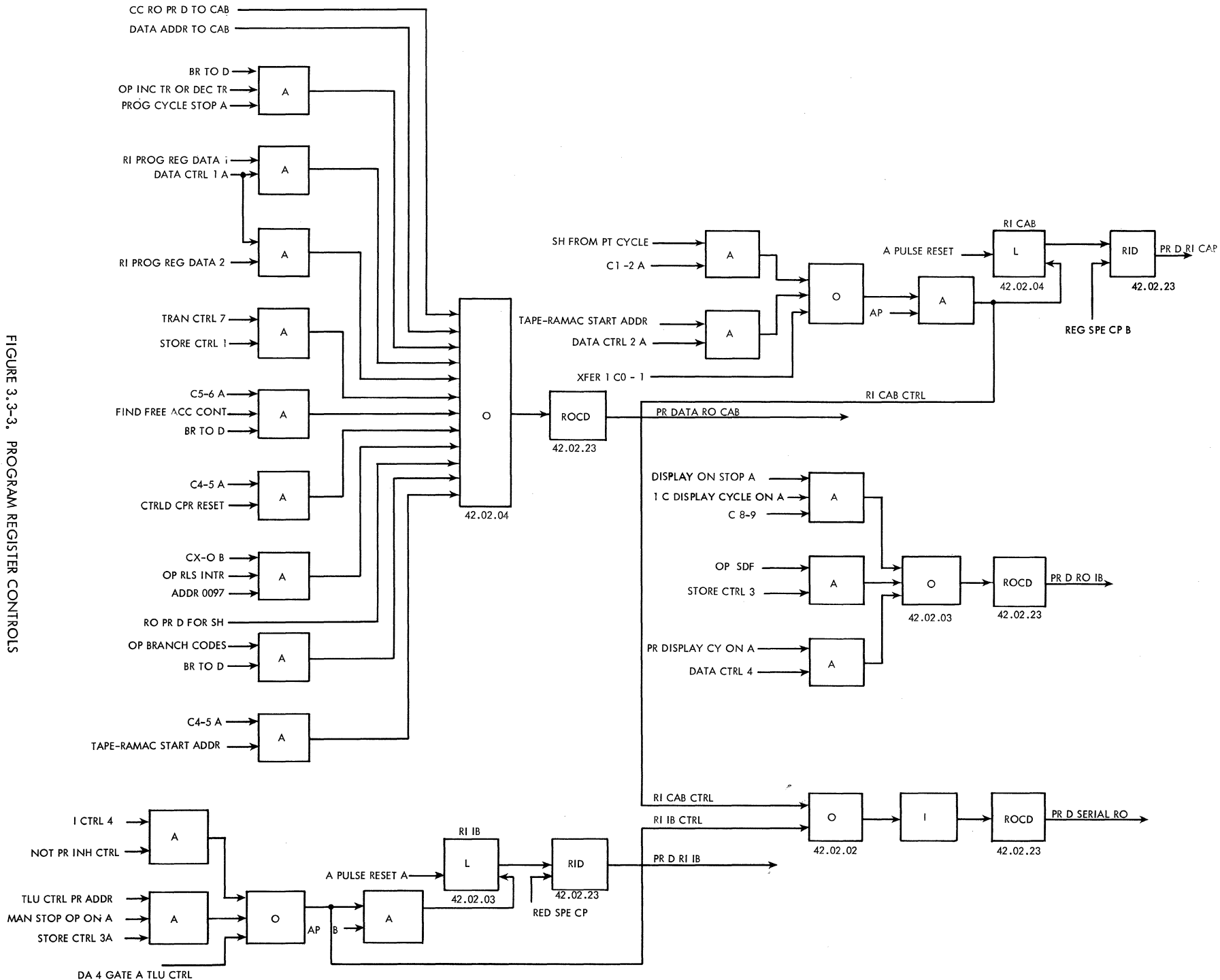


FIGURE 3.3-1. PROGRAM REGISTER LATCHES

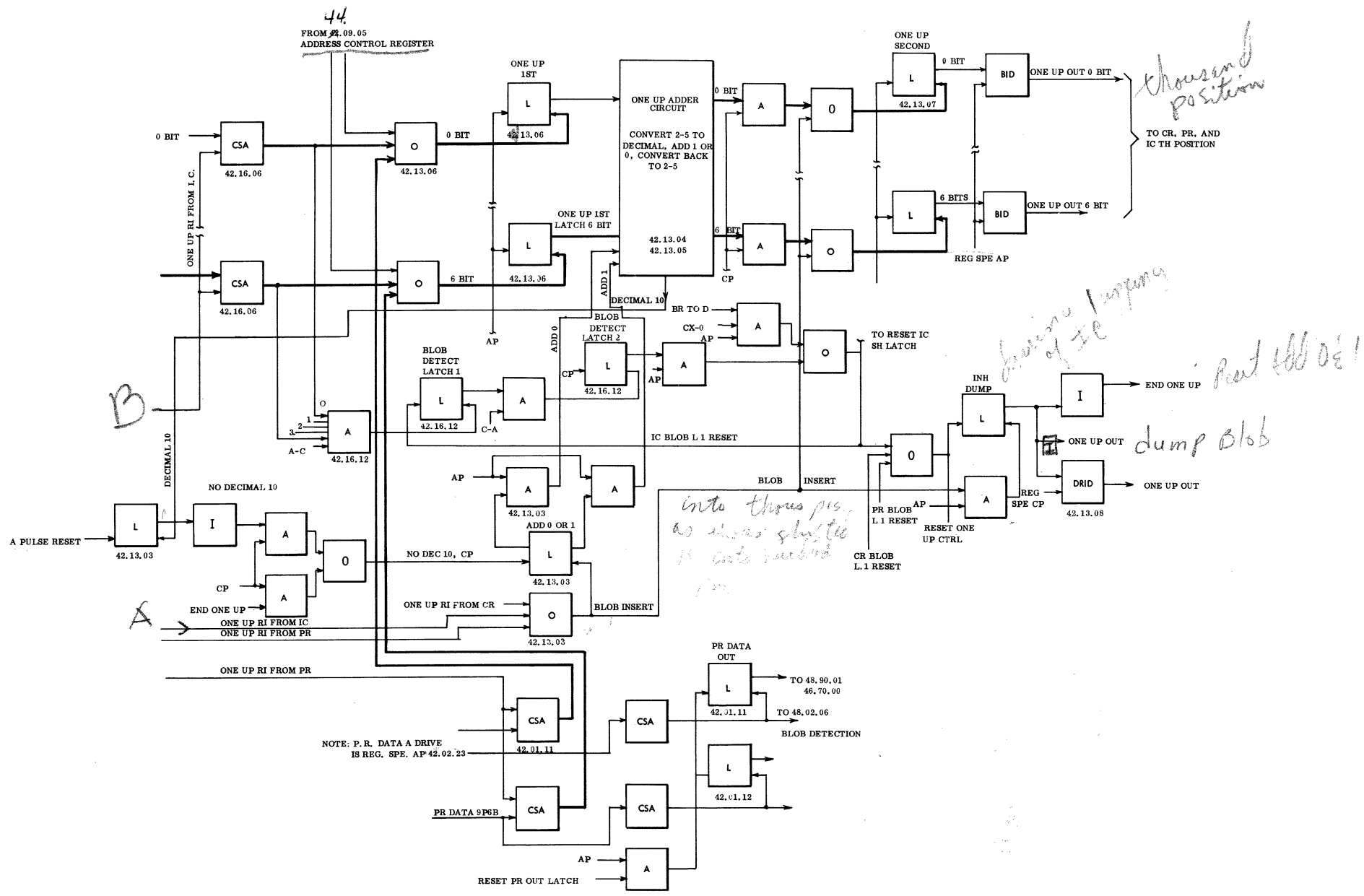
FIGURE 3.3-3. PROGRAM REGISTER CONTROLS



A

Timing chart p. 50

FIGURE 3.3-5. ONE-UP ADDER



4.0.00 PROGRAM EXECUTION

The IBM 7070 Program Cycle consists of three parts as follows (Figures 3.2-3 and 4.0-1):

Instruction - The next instruction to be performed is read out from core storage to the program register.

Index - If the program register positions 2 and 3 contain the location of an index word, the program register contents are added to positions 2-5 of the index word. If program register positions 2 and 3 do not contain the location of an index word (00), the index portion of the program cycle is skipped and the data portion is performed.

Data - If the operation code requires the use of the contents of the address specified by the data portion of the program register, the core storage location is read out to the arithmetic register except on index codes when it is read out to the auxiliary register.

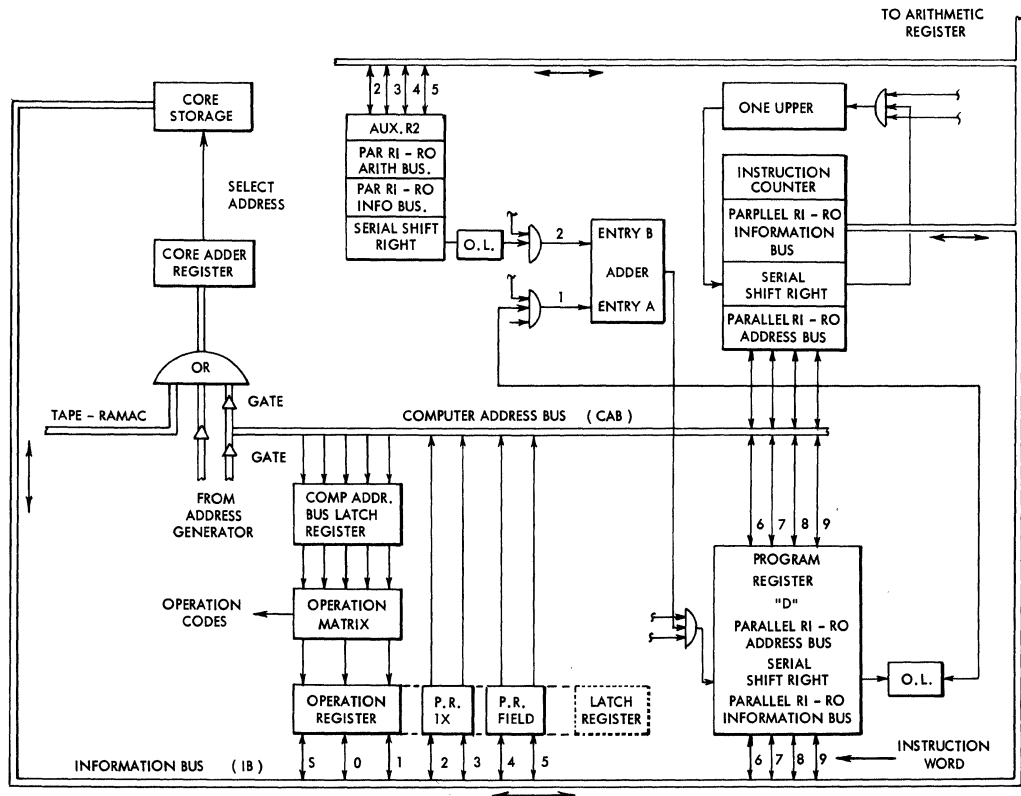


FIGURE 4.0-1. PROGRAM EXECUTION DATA FLOW

4.1.00 INSTRUCTION CYCLE

4.1.01 INTRODUCTION

Figure 3.2-3 shows the three portions of the 7070 program cycle along with the timing of the control rings which operate during each part of the cycle. While the 7070 is running, the instruction latch is turned on by an end-of-data operation signal. The instruction latch output then initiates the following:

1. A request to core storage for access time.
2. After the memory request is acknowledged by a return signal, the address in the instruction counter is transferred back to the core storage control circuits in order to select the next instruction.
3. The next instruction to be performed is read from the selected storage location into the program register.
4. The address in the instruction counter is increased by one.

These various operations are timed by the steps on the instruction control and program rings shown in Figure 3.2-3.

A summary of the 7070 basic program cycle, for codes requiring initial access to data in core storage is shown in Figure 4.1-1. Positive logic circuit summaries of the basic program cycle are shown in Figures 3.3-2 and 4.1-2.

4.1.02 DATA FLOW AND CIRCUITS

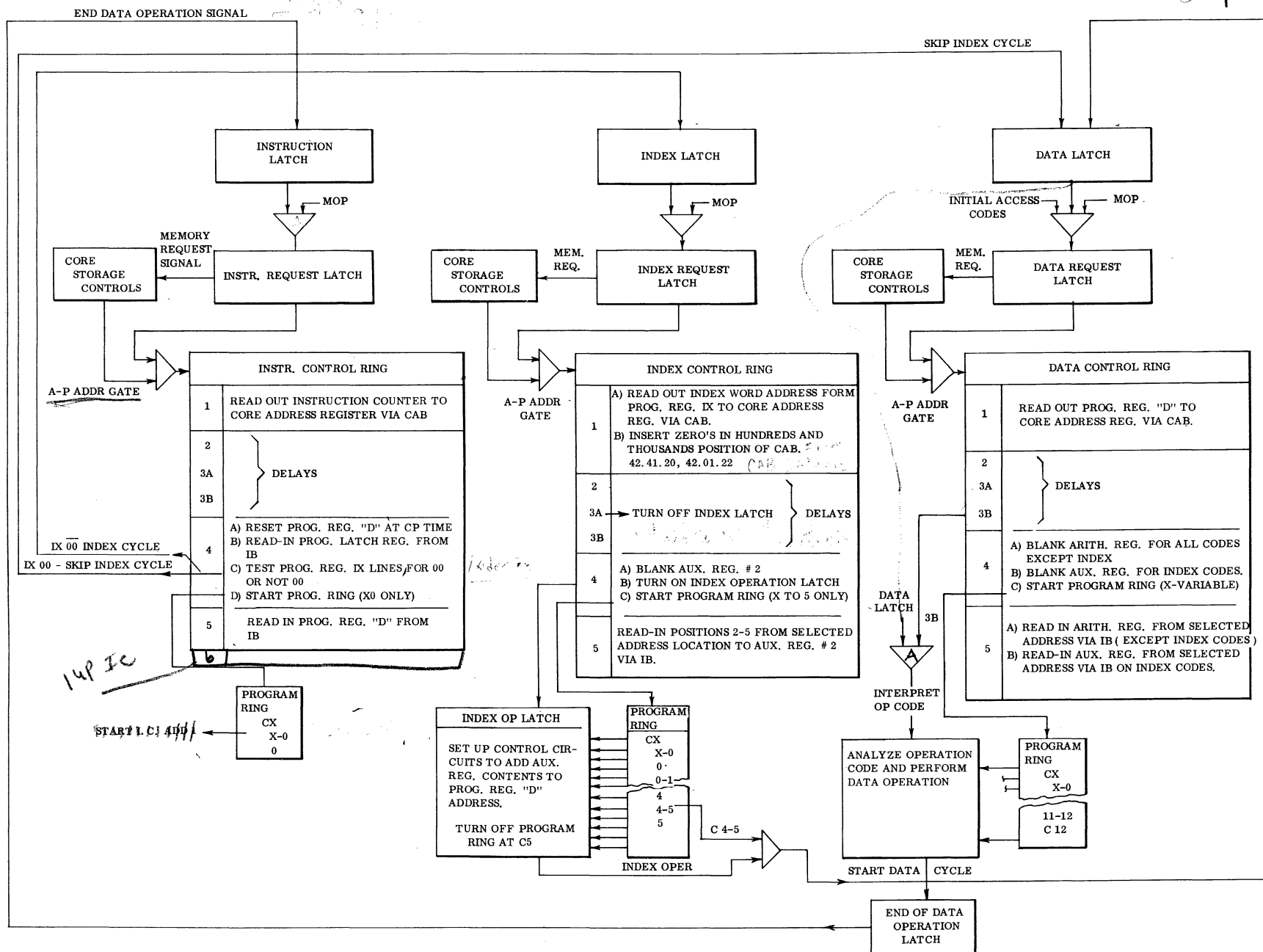
Instruction Request

During the following discussion, continued reference should be made to Figure 4.1-1 and the sequence chart of the instruction cycle (Figure 4.1-3).

When an end-data operation or other signal occurs to signal that an instruction cycle should begin, the instruction latch is turned on. The instruction latch output then switches with a memory out pulse (MOP) to turn on the instruction request latch. The MOP signal is developed by switching a memory V-pulse with an A & P D-pulse (see logic 42.40.00). This synchronizes the resulting memory request gate with the 6 usec core storage input-output cycles. When the core storage address selection control circuits are ready to receive the address of the instruction counter, an A & P address gate is returned to the program control circuits. The A & P address gate causes an instruction control ring, step 1, output and allows the ring to advance through step 5.

Read-Out of the Instruction Counter.

Figure 4.0-1 shows the data flow path from the instruction counter to the core address register via the computer address bus. The following circuits are necessary to read out the instruction counter contents in parallel to the computer address bus and to regenerate the contents of the instruction counter (IC).



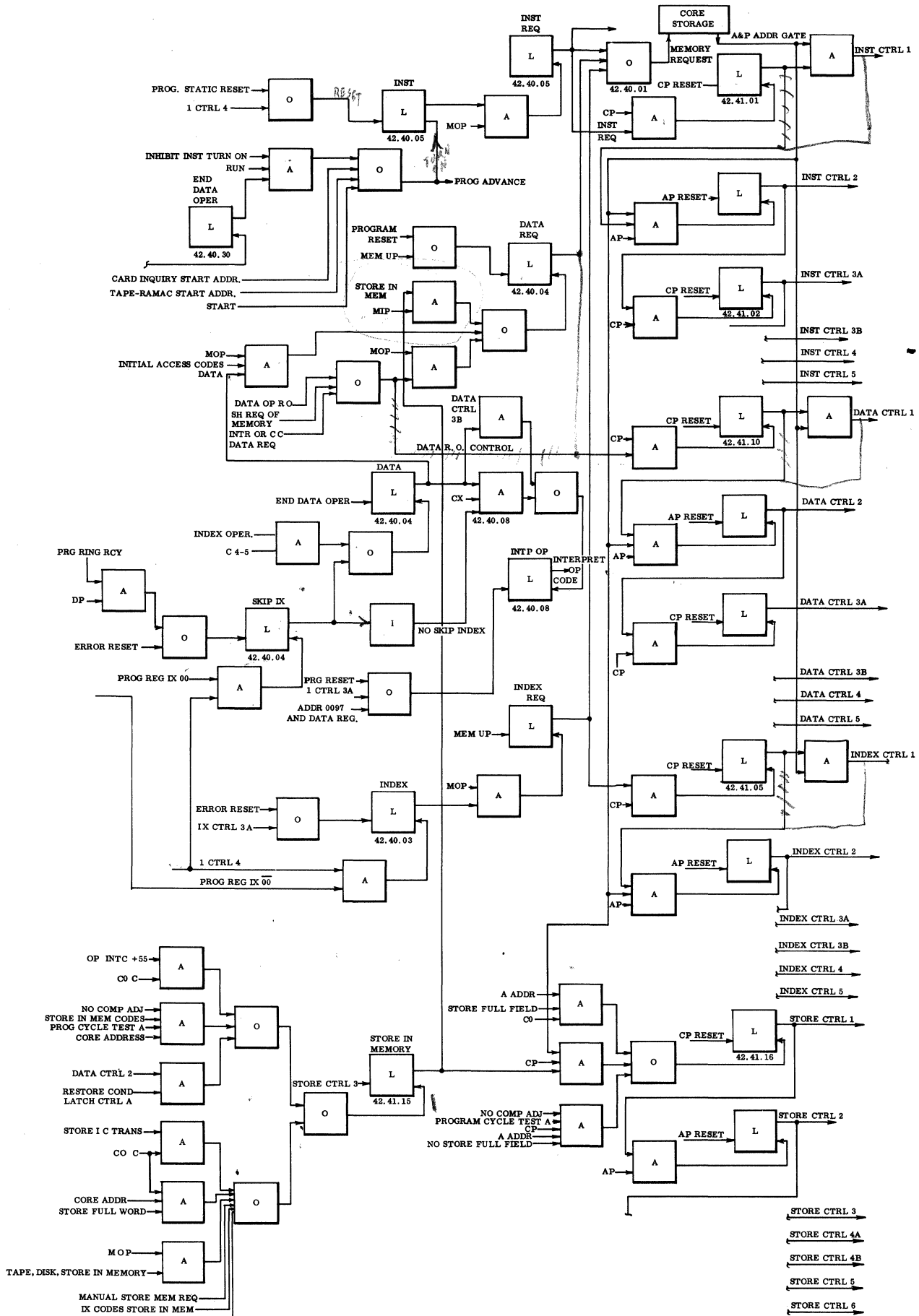


FIGURE 4.1-2. CONTROL RINGS

Objective: RO IC to CAB and regenerate.

<u>Gates Required</u>	<u>Timing</u>	<u>Key Gating Conditions</u>	<u>Circuit Location</u>
IC A Drive In	AP-RO	Reg Spe AP	42.16.17
IC Regen Drive	CP-RI	No IC Shift	Fig. 3.3-4
IC RO CAB Drive	C-C	I Ctrl 1	Fig. 3.3-4
IC Serial RO	Up	No RI IB or CAB	Fig. 3.3-4

The CAB storage capacitors are sampled to turn on address-trigger registers in core storage which select the address of the next program step. The contents of this storage location are then read out to the information bus capacitors at the beginning of the next memory-out cycle (Figure 4.1-3).

Read-in program register latches from information bus

Figure 3.3-1 shows how I Control 4 switches with (Not) PR inhibit control to provide an RI gate to set the PR latches (OP, IX, and Field) from the information bus capacitors.

Read in Program Register D from Information Bus

Program Register D must be reset prior to read-in of the address portion (Positions 6-9) of the new instruction. The reset is accomplished by permitting the read-out pulse (A-time) to flip the set positions and at the same time holding the serial ROCD cut off. This takes place during I Ctrl 4 time.

The IB storage capacitors are sampled at CP time to cause appropriate cores to be set in their 1 bit condition. Figure 4.1-3 shows the timing of the operation.

Objective: Reset Program Register and RI PR from IB

<u>Gates Required</u>	<u>Timing</u>	<u>Key Gating Conditions</u>	<u>Circuit Location</u>
PR A Drive In	AP-RO	Reg Spe AP	42.02.23
PR Data Serial RO	----	RI IB Ctrl up at I Ctrl 4	Fig. 3.3-3
PR Data RI IB	CP-RI	I Ctrl 4 (Not) PR Inh Ctrl	Fig. 3.3-3
PR Regen Drive	CP-RI	No Serial Advance	Fig. 3.3-2

Test Program Register IX Lines for 00 and Not 00

Figure 4.1-2 shows the circuits which test the PR IX 00 and not 00 lines to determine whether an index cycle is to take place or not. The IX 00 lines switch with I Control 4 to turn on the Skip IX latch. This in turn causes the data latch to go on and the index operation is skipped.

If the output of the PR IX latches is not 00, the index latch is turned on at I control 4 time and an index cycle results. This is in Section 4.2.00.

Start the Program Ring

The I control 4 gate switches with an A & P information gate (Figure 3.2-2) to turn on ring run. Either index or skip index cycle switches with program ring C0 to turn on ring stop and turn off the ring run latch. Thus, the program ring runs from CX through C0 on an instruction cycle.

Advance the Instruction Counter by One.

Data Flow - At the end of the instruction cycle, the CX-0 output of the program ring initiates the addition of 1 to the units position of the instruction counter. As shown in Figure 3.3-4, 5, the instruction counter contents can be shifted to the right through the one-up-adder circuit. The one-up-adder output is then directed back to the thousands position.

One-Up Adder Circuit - Logics 42.13.04 and 42.13.05 show the one-up adder circuit. Note that the 2-5 serial IC output is converted to decimal form. One or zero is added to the decimal representation, and the sum is converted back to a two-out-of-five form. If the sum of the addition is a decimal 10, a carry to the next higher order is indicated, and the add 1 circuit is kept on for an additional shift cycle.

Blob Insert and Detection - Figure 3.3-5 shows how all five one-up-second latches can be forced on by the Blob, Insert circuit. Thus a blob, or marker, is inserted into all five thousands position cores as the thousands bits are shifted into the hundreds position cores. The IC serial output from the circuits position is continually checked by the blob detection circuit and the operation is stopped when the blob has shifted out of the units position.

Control Circuits for IC Advance (Figure 4.1-4) - Objectives: Shift IC to Right

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
IC Shift Latch	On at CX-0, AP	CX-0, Skip IX or Index	Figure 3.3-4
IC Serial RI Drive	IC SHL, CP	IC SH Latch	Figure 3.3-4
Drop IC Regen Drive	----	IC SH Latch inverted	Figure 3.3-4
IC A Drive In	Reg Spe AP	----	42.16.17

Gate IC Serial Output to One Up Adder (Figure 3.3-4 & 5). The CX-0 gate which turns on the IC shift latch also switches with an AP to turn on the one-up-control IC latch. The output of this latch gates the serial IC output to the one-up-adder circuit.

Insert Blob into the Thousands. The CX-0 gate also forces on all five one-up-second latches, to form the one-up-out (AP) input to the thousands position. Thus, the blob is inserted into the thousands position as the thousands bits are shifted into the hundreds position.

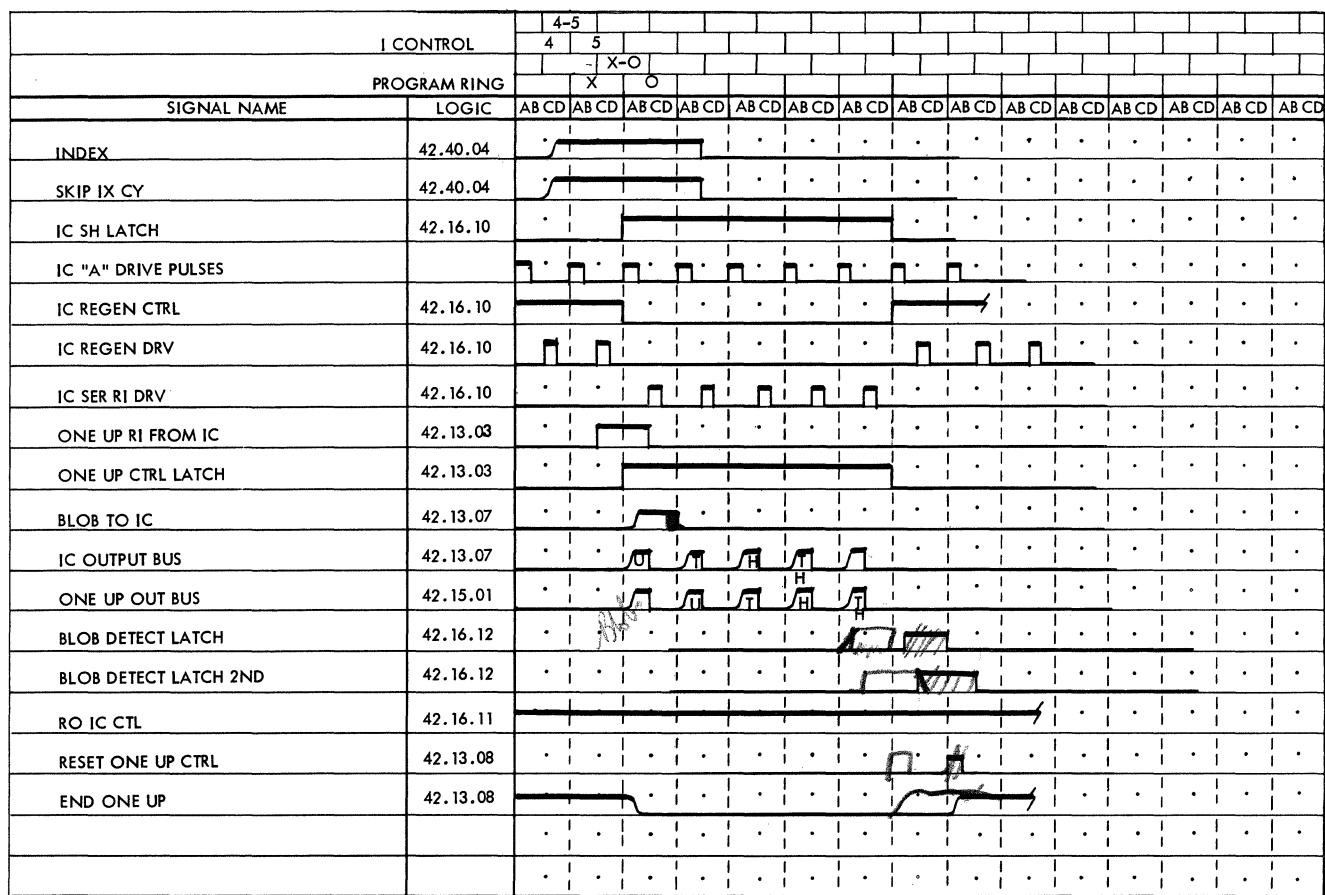


FIGURE 4.1-4. INSTRUCTION COUNTER CONTROL

Add 1 to Units Position. The CX-0 (Blob Insert) gate also turns the Add 0 or 1 latch to the Add 1 position. Its output switches with an AP to provide an Add 1 pulse to the one-up adder circuit. If the units position is 9 so that a carry is needed, a Decimal 10 output from the one-up adder circuit prevents the Add 0 or 1 latch from being turned to its Add 0 position. If the address in the IC is 9999 before the updating of the counter begins, a No Decimal Ten signal does not occur to turn the Add 0 or 1 latch back to Add 0. In this case, an end-one-up signal switches with a CP to turn the latch back to Add 0.

Sense for Blob - The serial output of the IC is continually checked for the 5 bit blob as shown on Figure 3.3-5. The blob detection is delayed by the blob detect 2 latch. The blob detect 2 latch output is used to reset the following latches:

One Up Control IC
 Blob Detect 1
 IC Shift
 Inhibit Dump

The resetting of the IC controls allows regeneration of the cores to begin.

4.2.00 INDEX CYCLE

4.2.01 INTRODUCTION (Figure 4.0-1 and 4.1-1)

During instruction control ring 4 time, a test is made of the IX (Index) field of the new instruction to determine whether the instruction is to be indexed or not. If the IX register is 00, no indexing is to take place and the program cycling proceeds directly to the data portion. If the IX register is not 00, the index latch comes on to signal the beginning of the indexing portion of the cycle. During the index operation, the contents of the program register are modified by positions 2-5 of the core storage location specified by the IX part of the instruction. The index latch output initiates the following operations:

1. A request to core storage for access time.
2. After the request to core storage is acknowledged by a return signal, the index word address is transferred from the program register IX positions to the core storage address selection control circuits. During the transfer, zeros are inserted in the hundreds and thousands positions to put the index word address in the form 00XX.
3. Positions 2-5 of the selected index word are transferred from core storage to the auxiliary register.
4. The program ring is started during step 4 of the index control ring. The function of the program ring is to time the serial addition of the auxiliary register contents to the program register D contents.
5. The index operation latch is turned on to control the duration of the program ring and to control the turn-on of the gates in the data flow path from auxiliary register to adder, from program register D to adder, and from adder to program register.
6. At the end of the addition process, the program ring output signals the end of the index operation and the data portion of the cycle begins.

4.2.02 DATA FLOW AND CIRCUITS

Index Request (Figures 4.1-1 and 4.2-1)

If the contents of the program register IX latches is not 00, an instruction control 4 gate causes the index latch to turn on (Figure 4.1-2). The next following memory-out pulse (MOP) causes the index request latch to turn on. Index request results in a memory request in the same manner as instruction request. As soon as core storage is clear to make the address selection, an A & P address gate is returned to the program control circuits. This causes step 1 of the index control ring to turn on and allows the ring to advance.

Transfer IX Address to the Core Address Register via CAB

Figure 3.3-1 shows how the program register IX latch output is sampled at IX control 1 time to cause an A pulse driver operation to insert a bit on the computer address bus. Also shown on Figure 3.3-1 are the circuits necessary to insert zeros in the hundreds and thousands position of the CAB lines.

Reset auxiliary 2 and RI positions 2-5 of the core address location via information bus.

Objective: RI Auxiliary 2 from IB

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
Aux 2 RI IB	Aux 2 RI, IB, CP	Index, IX Ctrl 4, AP	Figure 4.2-2
Block Aux 2 Ser RO	----	Index Op, IX Ctrl 4 inverted	Figure 4.2-2
Aux 2 Regen Ctrl	----	No Adv Aux 2	Figure 4.2-2
Aux A Drive In	Reg Spe AP	Reg Spe AP	42.31.15

Because serial RO is down, the regen capacitor cannot be charged and the Aux 2 regen drive CP does not regenerate the bit. If the IB bus capacitor is charged, the Aux 2 RI IB CP causes the read-in of a bit. At the end of this operation, the auxiliary register contains the address modifier.

Turn on the index operation latch, logic 42.40.09.

The index operation latch is turned on by an index control 4 gate. The latch controls the addition of the program register contents to the auxiliary register.

Start and Stop the Program Ring.

Index control 4 switches with an A & P information gate from core storage (Figure 3.2-2) to turn on ring run. The C5 program ring output switches with index operation to stop the ring and turn off ring run. While the ring is on, the contents of Aux 2 are added to the program register and the result is returned to the program register.

Add Aux 2 to the program register.

Data Flow. Figure 4.2-2 shows the data flow elements necessary to add Aux. 2 to the PR. In addition to the gates shown, the PR data RI adder gate (Figure 3.3-2) must be on to allow the serial adder output to shift into the program register.

Serially RO Aux. 2 to the Adder - Objective: RO Aux. 2.

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
Advance Aux 2	Adv Aux 2 Reg Spe CP	Index Op, C0-1	Figure 4.2-2
Drop Aux 2 Reg Ctrl	-----	Adv Aux 2 inverted	Figure 4.2-2
Aux A Drive In	Reg Spe AP	Reg Spe AP	42.31.15
RO Aux 2, Ch2	On C0-1 AP Off C4-5, AP	Index Op	Figure 4.2-2
True Add or Comp add Gates		See Adder Controls	44.40.06

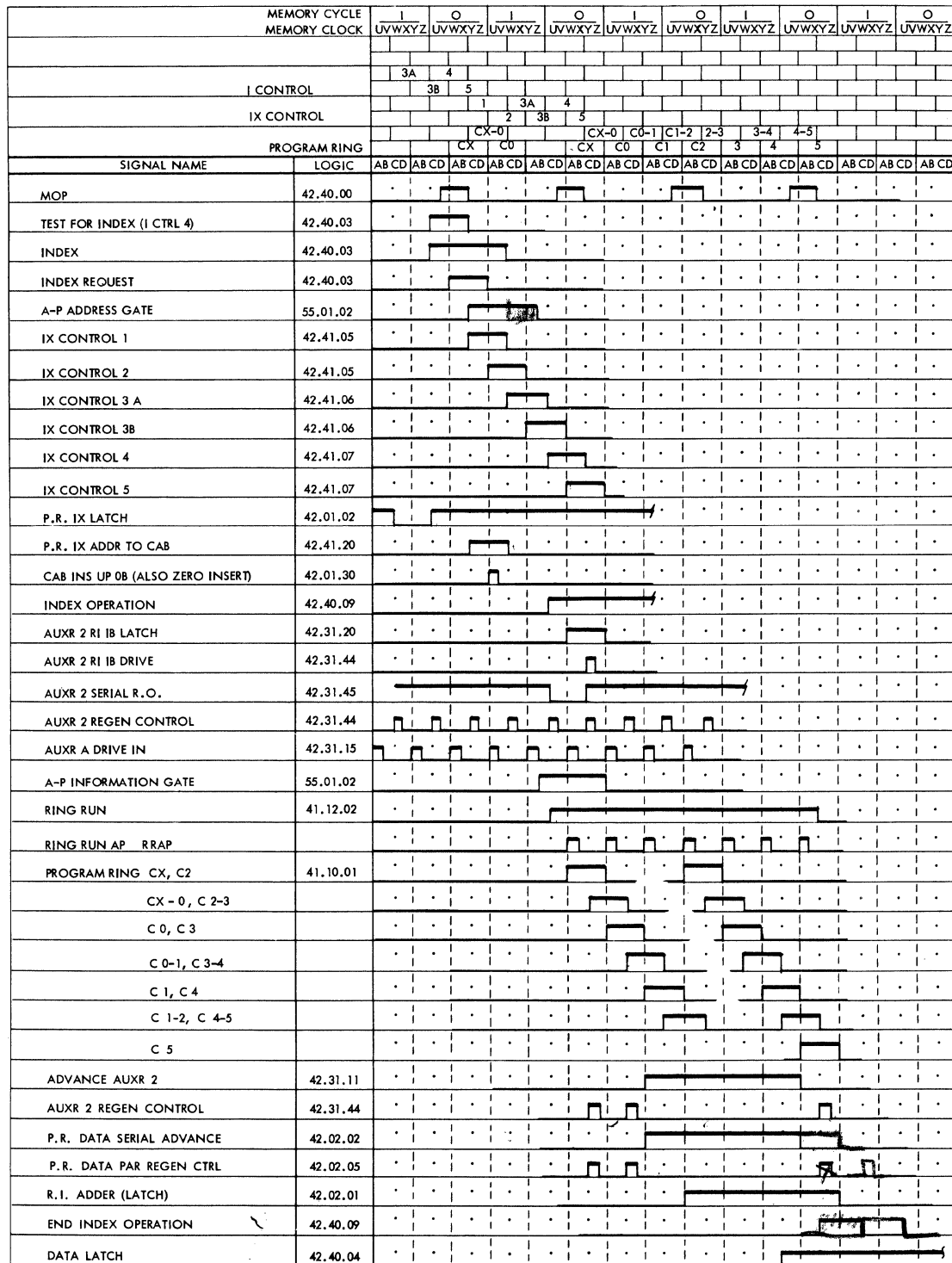
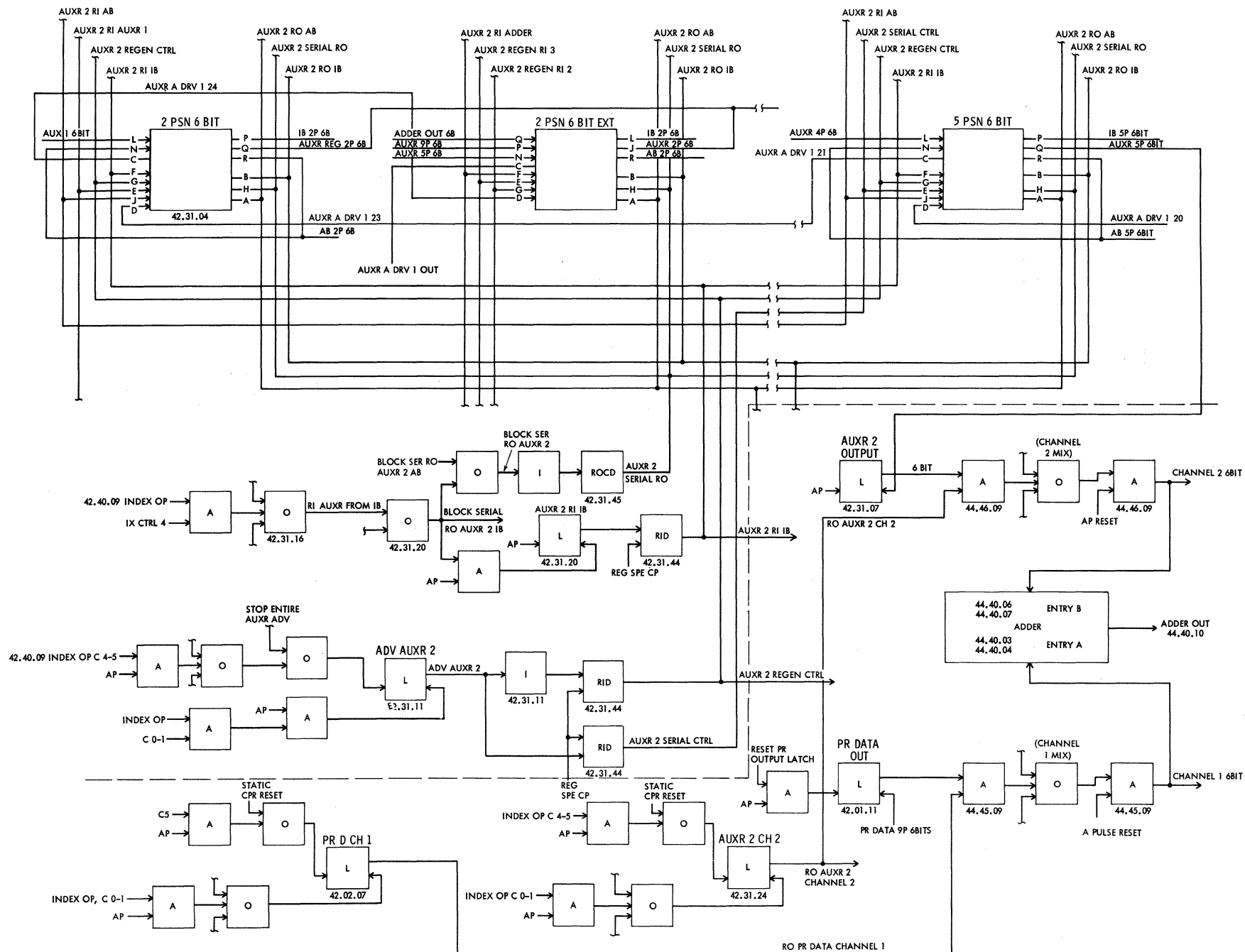


FIGURE 4.2-1. INDEX CYCLE SEQUENCE

[illegible]

Objective: Serially RO Program Register to the Adder.

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
PR Serial Advance	PR Ser Adv Reg Spe CP	Index Op,C0-1,AP	Figure 3.3-2
Drop PR Regen Ctrl	-----	Serial Advance inverted	Figure 3.3-2
PR Data A Drive	Reg Spe AP	Reg Spe AP	42.02.23
RO PR Data Ch 1	On C0-1, AP Index Op Off C5, AP		Figure 4.2-2

Objective: Read Adder Output into PR Thousands Position

The RI Adder latch (Figure 3.3-2) is turned on at C2, index Op time and turned off by an end-index operation, AP. The end-index operation signal is developed at C5, AP time when the index operation latch goes off (Logic 42.40.09).

Signal the Start of the Data Cycle

Index operation switches with C4-5 (Figure 4.1-2) to turn on the data latch to begin the data cycle.

4.3.00 DATA CYCLE

4.3.01 INTRODUCTION Figures 4.0-1 and 4.1-1

The operation register output is analyzed to determine the nature of the next operation to be performed. If the operation code involves the read out of the core storage location specified by the address in the D portion of the program register, the following sequence of operation is initiated by the data latch:

1. A request to core storage for access time.
2. When the request to core storage is acknowledged by the return signal, the contents of program register D is transferred to the core storage address selection control circuits.
3. The contents of the selected storage address are transferred as follows:
 - a. To the arithmetic register on all but index codes
 - b. To the auxiliary register on index codes
4. The program ring is started during data control ring 4 time. The program ring times any serial operation, such as addition, which takes place during the performance of the operation. The program ring runs a variable length of time during the data part of the cycle.
5. At the completion of the data operation, an end-of-data operation signal causes the instruction latch to turn on and the basic program ring cycle is repeated.

4.3.02 CIRCUITS (Figure 4.3-1)

Data Request

The data latch (Figure 4.1-2) can be turned on either by a skip index gate or by a C4-5 pulse at the end of an index operation. When the next MOP pulse arrives, the data request latch is turned on. This results in a memory request. When the core storage unit is ready to receive the address of the next instruction from the PR, an A & P address gate is returned to cause a Data Ctrl 1 gate and to advance the data control ring.

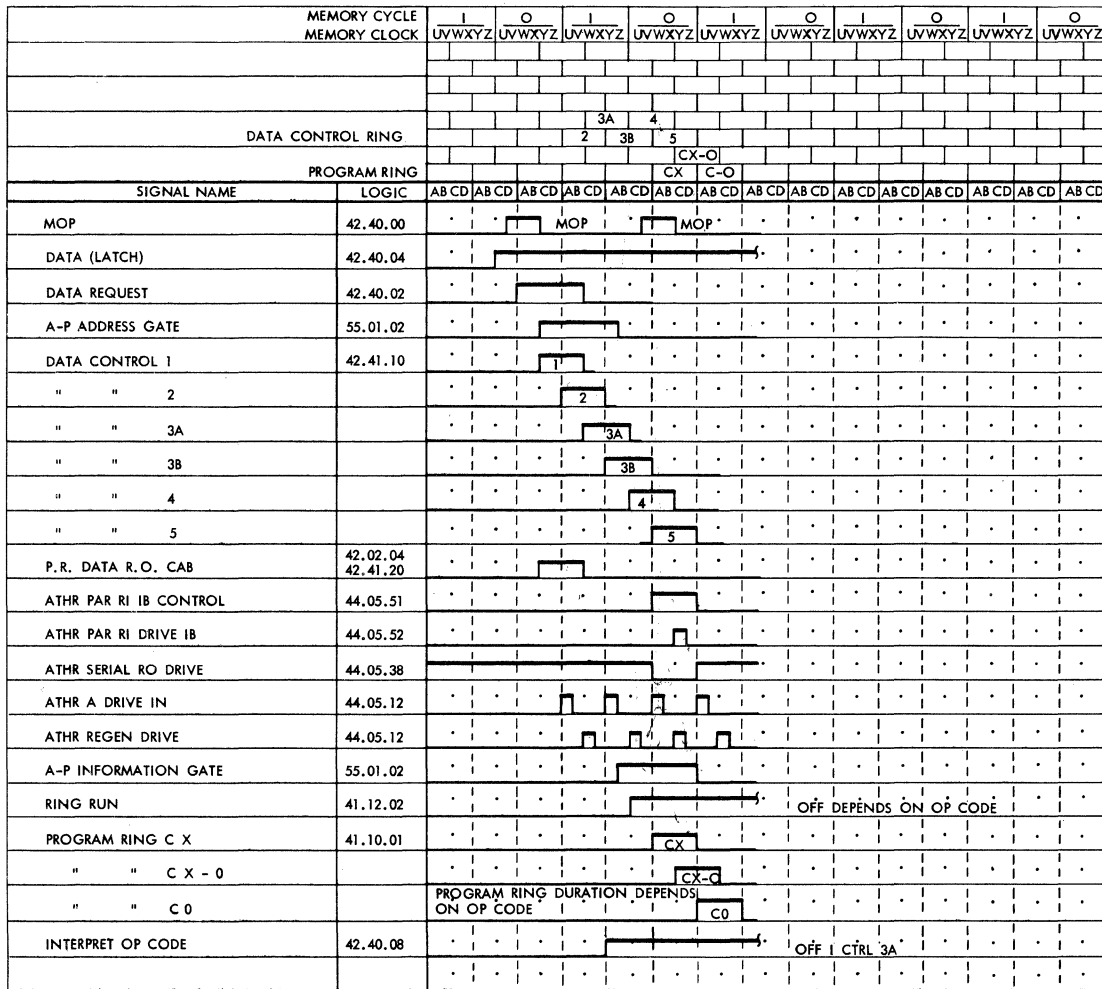


FIGURE 4.3-1. DATA CYCLE SEQUENCE

Read-out program register D to core address register via the Computer Address Bus (CAB)

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
PR Data RO CAB	Data Ctrl 1	Use Data Address Op Codes	Figure 3.3-3

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
PR Serial RO	---	RI Controls down	Figure 3.3-3
PR Data Par Regen Control	Reg Spe CP	Serial Adv down	Figure 3.3-2
PR A Drive In	Reg Spe AP	Reg Spe AP	42.03.23

This operation causes the regeneration of the bits in the PR and the read out of the bits to the core address register via the CAB.

Reset and Read-In Arithmetic Register from IB for All but Index Codes.

(The Arithmetic Register is shown in Logics 44.01.XX)

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
Athr Par RI IB	On Off	D Ctrl 4 AP, AP	Athr Codes, Core Address
Drop Athr Serial RO Drive	----	Athr Par RIIB Up	44.05.38
Athr Regen. Drive	Reg Spe CP	None shown	44.05.12
Athr A Drive In	Reg Spe AP	Reg Spe AP	44.05.12

Dropping of arithmetic register serial RO causes reset of the register. The data to be used in performing the operation code is in the arithmetic register at the end of the read-in.

Reset and Read-In Auxiliary Register for Index Codes

(The Auxiliary Register is shown in Logics 42.31.XX.)

<u>Gates Required</u>	<u>Timing</u>	<u>Gating Conditions</u>	<u>Circuit Location</u>
Aux 2 RI IB	Data Ctrl 4, Reg Spe CP	Pre Sense IX Codes	42.31.18, 42.31.20
Drop Aux 2 Serial RO	Data Ctrl 4	Invert RI Aux from IB	42.31.20, 42.31.45
Aux 2 Regen Ctrl	-----	No Adv Aux 2	42.31.16, 42.31.44
Aux A Drive In	Reg Spe AP	Reg Spe AP	42.31.15

The Index Code operation is discussed in detail in a later section.

Control the Program ring

The program ring is started by switching Data Control 4 with an A & P information gate and card control inhibit data (Figure 3.2-2). It may be recycled or stopped in a variety of ways -- depending upon the operation code to be performed. Each case is discussed individually in the section on Operation Codes.

Interpret the Operation Code

The interpret operation code signal is turned on at data control 3B time by switching with the output of the data latch (Figure 4.1-1). It is turned on earlier if an index cycle is not performed (Logic 42.40.08). The development of specific operation codes is discussed later. In general the Op Matrix operates as follows:

1. The program register tens position operation code latch outputs are switched together to provide Op 0X, 1X, 2X, 3X.....9X signals (see Logics 42.11.20, 21).
2. The program register units position operation code latch outputs are switched together to provide Op X0, X1, X2.....X9 signals (see Logics 42.11.22, 23).
3. Signals developed as in 1 and 2 just described are switched with Op Register + or - signals to provide pre sense op interpret signals prior to the development of interpret op code during the data cycle (see Logics 42.10.XX). Note that signals such as No Access Code, Use Data Addr Mix, Op Data Access, Pre-sense IX Codes, etc., are developed to aid in gating the correct data to the A & P unit for use during the actual data operation. The actual OP code signal is not developed until interpret op code is switched with the outputs of 1 and 2 just described.
4. The program register sign position latch outputs are switched with interpret op code to provide Op +, Op - and Op + or - signals (see Logics 42.11.01 and 42.11.07)
5. Signals developed as in 1 and 2 are switched with 4 to provide the actual Op code signals which cause the performance of the indicated operation.

Signal End of Data Operation

The End-Data Operation signal is developed in a variety of ways, depending upon the operation code (see Logics 42.40.30 and 42.40.40). As shown on Figure 4.1-1, the end-data operation signal turns off the data latch and turns on the instruction latch.

4.4.00 FIELD CONTROL

The normal word length of ten digits and sign frequently exceeds the requirements for factor sizes in an application. When the factors are small enough, two or more such factors can be stored in the same address location. A system of field control allows selecting a digit or a group of sequential digits from the word. The remaining digits are excluded from the computation. This feature allows increased factor storage and a reduction in handling time through the serial adder.

Digit positions four and five of instruction words for arithmetic operations indicate the size and location of the desired factor within the data word. Position four specifies the high-order digit, while position five indicates the low order. A field coding of 09 indicates the full word is to be used. A smaller field is selected by changing one or both field positions (03, 27, 59, etc.). A single digit can be selected by use of the same value in both positions of the field coding (00, 55, 99, etc.). A reversal of the high and low order digits in the field code indicates a program error (90, 65, etc.).

One limitation placed on the storage of multiple factors in a single word is the availability of only one sign position. This sign can be used with any portion of the word and on occasion for all parts. Many factors entering a computation are always positive and thus do not require a sign. Program controls permit the use of a factor with or without the word sign and the storage of a factor with or without sign consideration.

The designated factor positions are selected by using a field digit ring to scan the data word serially. Position five of the field register sets the field ring starting point. The ring position as it advances each digit time is matched with position four of the field register to determine the stop point of the scan shift. A check is made on the digit count of the transfer. Position five of the field register is 9's complement-added to position four by use of the adder. A carry on this operation indicates a reversal of digits and a program error. The one digit result is stored as a 10's complement in the latches of the field length register. The output of this register is switched with the program ring and the output checked for coincidence with the stop point of the field ring.

The movement of a factor from its designated position to the low order or the reverse is controlled by a combination of the field ring and the program ring. These digit rings serially scan the factor in and out of the arithmetic register. The move is accomplished by passing the data through the adder and returning it one digit later to the alternate area. This operation is combined, where possible, with a basic add operation by entering the second factor to the adder instead of channel zeros.

The adder cycle can be shortened from its basic ten digits, when both factors are smaller. The field register indicates the size of the data factor, while the significant digit scanner checks the size of the operated accumulator factor. The factor size as determined by the significant digit scanner is read into the shift register. As the transfer progresses, the shift register is shifted down, digit by digit, as the operated accumulator factor, now in the auxiliary register, is transferred out. When the shift register reaches A LAST, it signals the end of the factor. Indications of the end of both factors causes the adder operation to stop. If the transfer is to core storage, the field ring alone stops the adder operation, and an error is indicated if the operated accumulator factor is larger.

As a further check on the validity of the significant digit scanner operation, the auxiliary register is read out in parallel to the arithmetic bus and the significant scanner. Because the auxiliary register was not serially regenerated, but zero inserted from the high-order end, it should contain all zeros.

On complement-add operations, where the sign is reversed, the total must be re-complemented. When the result is to be transferred to core storage, the field register again sets the field ring for both read-out and read-in. If the total goes to

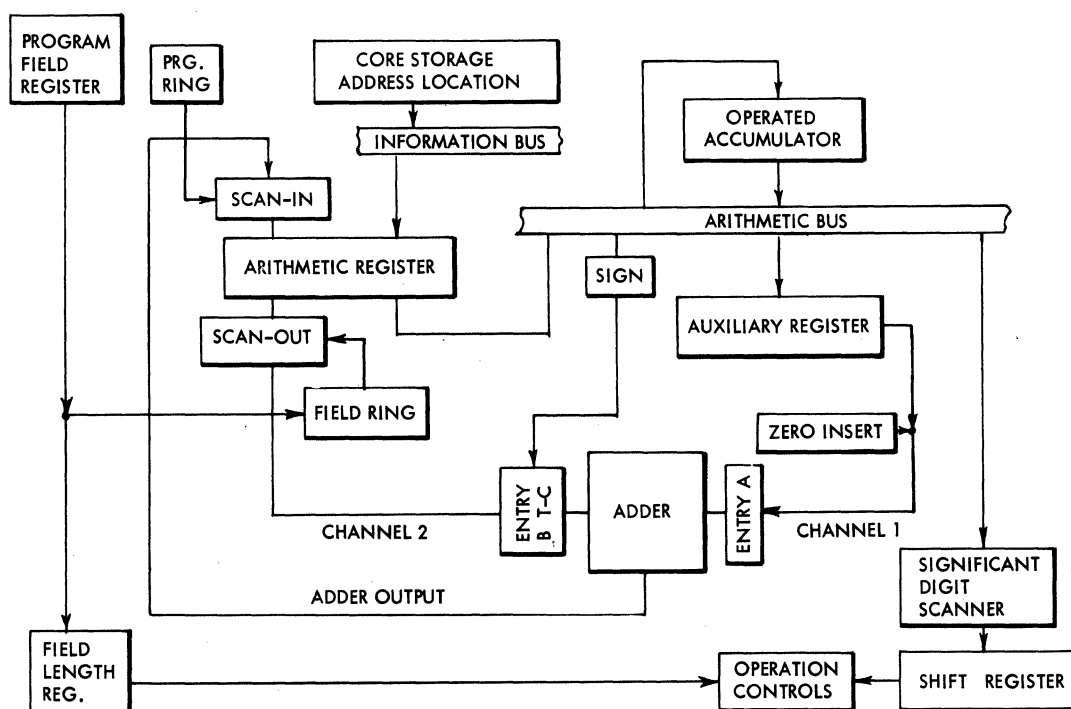
the operated accumulator, the program is reset to *9. The * value is determined by the stop point of the program ring from the preceding complement-add operation. The arithmetic register zero latches parallel insert zeros to the unused high-order positions to validate the word.

Specific applications of field control to the various operations is covered in more detail in the specific operation descriptions.

5.1.00 ADD - SUBTRACT & STORE

The data address of the instruction can be any core storage location or one of the three accumulators. The operation code of the instruction indicates the operation and accumulator to be used as the second factor. The general data flow for the add operations is discussed in two groups dependent on the result location. The one word transfer operations of zero and Add and Store Accumulator are also included because of their resemblance to add operations, when using a field control of less than full word.

With a data address located in core storage, the factor word is transferred in parallel over the information bus to the arithmetic register. Simultaneously, the operated accumulator is read-out in parallel to the arithmetic bus and regenerated. The arithmetic bus is sensed by the significant digit scanner for the location of the high-order digit during the transfer. It is then read in parallel to the auxiliary register. The operated accumulator sign latches are set from the bus, while the factor is transferred. The arithmetic register sign position is transferred to the arithmetic bus during the following digit time to set the address sign latches. If the data address is an



61

accumulator, the sign and data are parallel-transferred over the arithmetic bus during the second digit time. The two sets of sign latches and the operation sign are used to control the true or complement conditions.

The actual adding starts with the third digit time. The low-order position of the specified field in the arithmetic register is serially scanned by the field ring to adder entry B on Channel 2. At the same time the low order of the auxiliary register is serially shifted to read onto Channel 1 to Adder Entry A. The adder output is scanned serially into the arithmetic register by the program ring starting with the low-order register position. The adding continues until the field ring and the shift register indicate the end of both factors and the possible carry has been satisfied. If the operated accumulator factor is the larger, zeros are inserted on channel 2 to fill out the cycle. With a small operated accumulator factor, the transfer continues with the zeros from the auxiliary register. At the end of the adder cycle, the remaining high-order positions of the arithmetic register are filled with zeros by the zero latches, which were not reset.

The auxiliary register is parallel-transferred to the arithmetic bus and the significant digit scanner to check that no significant digits have been left. The register was not serially regenerated during the read-out. Zeros were inserted at the high order end. Finally, the arithmetic register is parallel-transferred to the operated accumulator over the arithmetic bus. The previous data in the accumulator is blanked.

Zero and Add to Accumulator

The zero and add operation differs slightly from the basic add to accumulator. The data address may be any core storage location or an accumulator. The data in the operated accumulator is ignored and only the data address factor is inserted. With a transfer of less than the full word, due to field control, the adder operation follows a basic add except that zeros are entered on channel 1 instead of the operated accumulator factor. The operation is always performed as a true-add with the sign being determined from the data address factor and the operation. When the arithmetic register is transferred to the operated accumulator, the blanking of the previous data effects the reset.

A zero and add operation with a field control of 09, or full word, eliminates the adder operation and transfers the arithmetic register directly to the operated accumulator.

Basic Add to Storage (Figure 5.1-2)

In an Add-to-Storage operation the data address location is being treated as an accumulator. The value of the operated accumulator is added to the value in the core storage location. Field control, in this operation, has the effect of limiting the size of the accumulating area, as well as that of the factor. If the operated accumulator factor is larger than the specified field or the result causes overflow, an error is indicated. A data address sign change is under control of a console switch to prevent accidental changes by the wrong field area. The factor in the operated accumulator is considered the operand and was entered on a previous operation or calculation.

With the data address located in core storage, the word is parallel-transferred to the arithmetic register over the information bus. Simultaneously, the operated accumulator is read out to the arithmetic bus in parallel and regenerated. The significant digit-

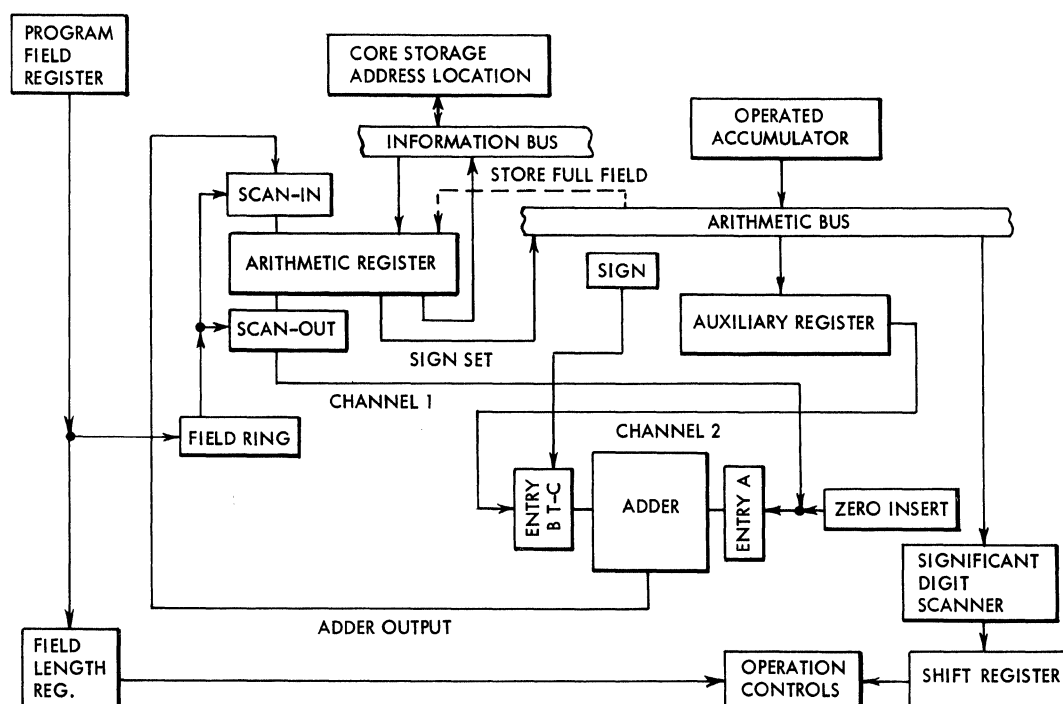


FIGURE 5.1-2. ADD TO STORAGE AND STORE DATA FLOW

scanner senses the arithmetic bus for the location of the high-order digit during the transfer. The factor is then read into the auxiliary register in parallel. The operated accumulator sign latches are set from the bus, while the factor is being transferred. The arithmetic register sign position is transferred to the arithmetic bus during the following digit time to set the address sign latches. If the data address is an accumulator, the sign and data are parallel-transferred over the arithmetic bus during the second digit time. The two sets of sign latches and the operation sign are used to control the true or complement conditions.

The actual adding starts with the third digit time. The low-order position of the specified field in the arithmetic register is serially scanned by the field ring to Adder Entry A on channel 1. At the same time the low order of the auxiliary register is serially shifted to read on to channel 2 to Adder Entry B. The adder output is scanned serially into the arithmetic register by the field ring starting with the low-order position of the specified field. The adding continues until the field ring indicates the end of the specified field. The shift register, as set by the significant digit scanner, is only used to detect an error condition if the operated accumulator factor is too large.

To complete the operation the store control requests a memory cycle and the data word is returned to its original location in core storage by parallel transfer on the information bus. If the original data address was an accumulator, the word is returned to that address over the arithmetic bus. The remainder of the data word is unchanged if the specified field contained less than ten digits. The value of the operated accumulator has not been changed and may be used for other operations.

Store Accumulator

The store accumulator operations differ slightly from the basic add to storage. The data address may be any core storage location or an accumulator. The original data in the specified field is ignored and the specified number of digits from the operated accumulator, starting with the low order, are inserted. When field control specifies less than full word, zeros are fed to the channel instead of the selected field of the arithmetic register. The operation is always performed as a true-add with a sign change treated in the same manner as the basic add to storage.

A store accumulator operation with a field control of 09, or full word, does not require the adder operation. The operated accumulator is parallel-transferred directly to the arithmetic register over the arithmetic bus. The accumulator was previously transferred to the auxiliary register and the significant digit scanner prior to the verification of the 09 field code. The sign of the operated accumulator is used in this case without question.

Sign Control

As previously indicated, add and subtract functions differ only in the use of the true or complement value of the operand. The true or complement controls are selected by analysis of the sign of the accumulator area, the sign of the operand, and the sign of the specified operation. The three signs are considered algebraically with the results shown in Figure 5.1-3. In add operations the sign of the result must also be determined. The accumulator area sign never changes with a true add operation. It only changes on complement add, when the operand factor is larger than the accumulator area. The latter case is covered in more detail under Recomplement. All sign changes are made in the arithmetic register and transferred to the accumulator area.

Two sets of sign latches are connected to sense the sign lines of the arithmetic bus. The operated accumulator sign latches are set during the transfer of the operated accumulator to the auxiliary register. The data address sign latches are set as a separate transfer of the sign position of the arithmetic register, when the location is in core storage. If the data address is an accumulator, the sign latches are set during the transfer over the Arithmetic Bus. On an add-to-accumulator operation, the operated accumu-

ADD - SUBTRACT SIGN CONTROL

Accumulator Area Factor A	Operand Factor O	Designated Operation	Add Cycle True or Complement	Result A > O	Result A < O
+	+	+	TA	+	+
+	+	-	CA	+	*-
+	-	+	CA	+	*-
+	-	-	TA	+	+
-	+	+	CA	-	*+
-	+	-	TA	-	-
-	-	+	TA	-	-
-	-	-	CA	-	*+

* Recomplement Sign Change

FIGURE 5.1-3. ADD--SUBTRACT SIGN CONTROL

lator sign is set in the arithmetic register after the data address sign has been read out. This is not necessary on add-to-storage operations because the data address becomes the accumulator.

The sign control for one-word transfer operations varies slightly in that the accumulator area sign is not used to determine the new sign and all adder operation is done with true-add controls. However, the data address sign is considered on a store-accumulator operation, when a sign change indicates an error.

In the use of absolute codes the sign of the operand factor is considered plus regardless of its value. In the case of the zero and add-subtract absolute, only the sign of the operation is used to set the accumulator. On store-absolute, the adder operation is true-add controlled but no signs are changed.

If either the data address or the operated accumulator sign is Alpha, the sign of the result is set Alpha for all normal add operations. The alpha sign rule is superseded in the case of absolute-add and store operations.

Recomplement

When the operand factor is larger than the accumulator area factor, an add operation using complement-add control ends with a complement result in the arithmetic register. This condition is detected by a no-carry output on the last adder digit. Re-complementing is accomplished by passing the result through the adder with complement-add control.

On an add-to-storage operation, the field ring is used both to read out of and to read into the arithmetic register. The field register remains at the instruction word setting, and thus, only the specified field is considered. The data is read out to channel 2 and Adder Entry B, while the Channel 1 digit insert feeds zeros to Adder Entry A.

The data flow is the same for the add-to-accumulator operation. The field ring is used to scan the data out of the arithmetic register, while the program ring scans it back. (The field register is set to an *9 value to scan the proper positions with the minimum of time. The * value in the tens position is determined by using the ~~10's~~ 11's complement of the program ring stop position from the add operation.)

The result is transferred to either the core storage location or the operated accumulator in the normal manner after the recomplementing. The sign is changed in the arithmetic register before transfer. The sign change to core storage may indicate an error, and therefore sets the memory sign change latch.

5.1.01 Add-Subtract Accumulator # (± 14 , ± 24 , ± 34) Add-Subtract Absolute Accumulator 1 (± 17)

All of the above codes follow a basic add-to-accumulator operation with variations in the sign control. The following major objectives are required to perform the operation:

- A. Data Cycle Start
- B. Data Read-In to Arithmetic Register
- C. Operated Accumulator Transfer to Auxiliary Register
- D. Set Field Control
- E. Operated Accumulator Factor Size Control

- F. Set Sign and True-Complement Latches
- G. Arithmetic Register Read-Out to Adder
- H. Auxiliary Register Read-Out to Adder
- I. Arithmetic Register Read-In from Adder Output
- J. Field End Controls
- K. Complement Adjust Cycle Setup
- L. Complement Adjust Cycle Adder Operation
- M. Complement Adjust Field End Controls
- N. Arithmetic Register Transfer to Operated Accumulator
- O. End Data Operation

A. Data Cycle Start

With the setting of the Data latch, a request is made of memory for a data word. When the request is recognized in core storage, an A & P address gate is developed which starts the data control ring. The ring reads out the data address to the computer address bus, setting its latches, and transfers to core storage. The latches are tested for a possible accumulator address, which, if found, inhibits the core storage read-out.

An interpret operation signal switches with the outputs of the Op Register latches to determine the presence of the add-to-accumulator code. To simplify over-all switching, the basic signal is mixed with other operation signals to form several gates used to set the pattern for the operation.

As the operation starts, the program ring is started to control the sequence and time of the various steps.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Data Latch	IX End	4D-42.40.04
Set Data Req Latch	MOP	5G-42.40.04
A & P Address Gate	MOP - XP	4E-55.01.02
A & P Info Gate	ZP	5D-55.01.04
Start Data Ctrl Ring	Addr Gate	5B-42.41.10
Set CAB Latch Reg	DC1 AP	4G-5F-42.05.01
999X Address	DC1	3H-42.05.05
Core Address	DC1	5D-44.05.50
Interpret Operation	DC3B	3C-4B-42.40.08
Op Add to A	DC3B	2G-3G-42.11.05
Op Add-Subt	DC3B	5B-42.11.05
Op AAB or SAB	DC3B	3E-42.11.05
Op Add-Subt to A	DC3B	3J-4J-42.11.31

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Arith Codes	DC3B	3E-42.11.06
Fld Ctrl Code	DC3B	5A-42.11.16
Op A 1	DC3B	2A-42.11.50
A Overflow Codes	DC3B	5H-42.11.31
Start Program Ring	DC4 AP	3A-4A-41.10.01
Prog Ring Start	DC4	4D-41.12.01
Prog Ring Run	DC4 AP	5B-41.12.02

B. Data Read-In to Arithmetic Register

The data word may originate in either core storage or one of the accumulators. The logic for the two areas varies completely. The address has already been sent to the core area during the A & P address gate. The selected word reads into the arithmetic register during the A & P information gate at Data Control 5.

An accumulator address causes the core storage read-out and the bus validity checks to be inhibited. The selected accumulator is read out to the arithmetic bus and into the arithmetic register. The transfer must occur at a later time (C0) due to the use of the arithmetic bus for the transfer of the operated accumulator.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Address		
Athr Par RI IB Ctrl	DC4	2E-44.05.51
Blank Ser RO (P0)	DC4	5A-44.05.38
Athr Par RI IB	C0 AP	4E-5E-44.05.51
999X Address		
Inhibit Access	DC2	4E-5E-42.40.01
Addr A RO to AB		
A (2) Addr Op	DC3B	4D-5E-42.05.06
A (2) AB RO Ctrl	DC4	3J-44.21.30

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Par RI AB Ctrl	CX-0	4C-44.05.50
Blank Ser RO (P0)	CX-0	5A-44.05.38
Athr Par RI AB	C0-AP	5E-5F-44.05.50

C. Operated Accumulator Transfer to Auxiliary Register

The operation code indicates the accumulator to be used by its 10's digit. The specified accumulator is read out to the arithmetic bus and regenerated in the normal manner.

The accumulator factor is read into the auxiliary register, which for this operation functions as a single ten-position register. Each section of the register is controlled from its own switching. The normal serial read-out for regeneration is blocked during the arithmetic bus read-in.

The arithmetic bus capacitor clear circuits are blocked during the transfer because cores are being set. The significant digit scanner is also read in from the bus during the transfer by sampling with capacitor sense amplifiers. The scanner operation is discussed later.

<u>Major Signal</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Accumulator 1 RO AB	DC4	2F-44.11.01
Block Ser RO Auxr 1 AB (2&3)	DC4	3C-42.31.17
Auxiliary RI from AB		
Auxr 1 RI AB	DC4 AP	5A-6A-42.31.17
Auxr 2 RI AB	DC4 AP	4D-5D-42.31.19
Auxr 3 RI AB	DC4 AP	4B-5B-42.31.21
AB Par Clear Ctrl	DC4 A - A	5H-42.31.21

D. Set Field Control

The field register is read in from the instruction word to denote the positions of the data word to be processed. The units position is used to start the field ring for read-out control. The ring progresses digit by digit until the ring position matches the tens position digit. A check is made on the digit count by use of the field length register. The units position of the field register is subtracted from the tens position with an adder operation. Use of the 9's complement value results in a carry if an illegal value is set. The adder output is the 10's complement of the digit count and is placed in the field length register bit latches.

Example:

1. A data word is field defined 36. This indicates that 4 digits positions are to be scanned out of the arithmetic register. The result of the 9's complement addition is the value 6 (6 and 0 bit) which is the 10's complement of 4.
2. A data word is field defined 18. This indicates that 8 digit positions are to be scanned out of the arithmetic register. The result of the complement add operation is the value 2 (2 and 0 bit) which is the 10's complement of 8.

The Error Check is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Reg to Adder (OPFCX)	CX	3B-44.02.23
RO Fld Reg TP CH 1	CX	3B-42.07.01
RO Fld Reg UP Ch 2	CX	3D-42.07.01
Complement-Add	CX	5E-44.41.15
No Carry Ins	CX AP	4C-44.41.02
RI Fld Size from Adder Output (AO)		
Fld Size Reg Reset	CX BP	2H-3H-44.02.20
RI Fld Size Reg	C0 BP	3E-4G-44.02.23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44.02.20
Fld Prg Error	C0 CP	3E-4E-44.02.23
Set Field Ring		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Ring Pos. (F 1 - 9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
Advance Field Ring		
On Time Latch (F 9)	AP	4C-5G-44.03.01
1/2 Time Latch (F 9 - 8)	CP	4F-5F-44.03.01

E. Operated Accumulator Factor Size Control

The significant digit scanner register is set with the high-order digit location during the transfer of the operated accumulator factor. The scanner immediately transfers the indication to the single-core shift register similar to a logic core extender. The entry is made with the shift right control rather than the regeneration read-in because the serial capacitor of the next higher-order position is used. The shift register is used as a ring to count out the digits during serial transfer of the factor to the adder. When the bit serially reads out of the low-order position, the significant digits have all been transferred. The Field End detection is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set SDS Check Latch	DC3	3G-4G-5G-44.06.05
SDS On	DC4 DP	3A-44.06.09
SDS Read-In (9P)	DC4 AP	3D-44.06.01
SDS RO to SCSR		
SDS RO Ctrl	CX-0 AP	4B-44.06.00
Shift Reg Sh Rt Ctrl	CX-0 AP	2B-4C-5B-44.50.18
Blank Sh Reg Par Regen	CX-0 AP	3B-44.50.19
SCSR Ser RO Ctrl	CX-0 CP	4F-44.50.19

F. Set Sign and True-Complement Latches

With a normal add-subtract operation the signs of both factors and the sign of the operation are matrixed to determine the actual function to be performed. The operated accumulator sign is sensed from the arithmetic bus during its transfer to the auxiliary register. The addressed sign is read in a like manner one digit time later if the address is an accumulator. With a core storage address, the sign alone is read out to the arithmetic bus to set the latches. To set the true and complement latches, an alpha sign is considered plus. The factor signs are compared for like or unlike and then switched with the operation sign to select the function. Like factor signs select the indicated op code function, while unlike factor signs reverse the functions. At the start of the adding cycle, the operated accumulator sign is inserted into the arithmetic register for the total.

An add or subtract absolute operation differs only in that the addressed factor sign plus latches are forced. The arithmetic register sign read-out to arithmetic bus and read-in arithmetic bus address sign signals are inhibited to prevent the use of the actual sign. The true-complement selection and the insertion of signs into the arithmetic register cores are the same as for the normal add operation.

The sign latches retain their information until the end of the operation to determine the sign of a complement-add operation. With numeric signs the operated accumulator sign is unchanged except on a complement-add operation requiring an adjust cycle as discussed later. If either of the factor signs is alpha, the accumulator sign is changed to alpha at the end of the adding cycle by insertion into the arithmetic register.

When complement adding, it is necessary to insert a carry on the first adder digit time to produce a 10's complement. To satisfy the adder input switching, a no-carry insert must be used with a true-add function. On successive digit times the adder develops its own carry and no-carry signals.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Operated Reg Sign Latches		
RI AB Op Sign	CX	3B-5A-42.11.30
Set Op Reg Sign Latches (+)	CX	5A-6A-44.35.02
Set Address Sign Latches		
Normal Add		
RI AB Addr Sign	C0	4C-42.11.31
Athr RO Sign to AB	CX-0	3B-42.11.31
Set Addr Sign Latch (+)	C0	5A-6A-44.35.03
Absolute Add		
Addr Sign Plus Ins	C0	3F-5F-42.11.31
Normal Addr Sign Inh	DC3B	3F-2C-42.11.31
Set Addr Sign Latch (+)	C0	3A-6A-44.35.03
RO Signs T - C	DC 3B	3B-42.11.10
Signs True-Add ++(+)	C0	3B-5B-44.35.04
Signs Comp-Add + (+-)	C0	3E-3F-44.35.04
Set T - C Latches		
True-Add Latch	C1	3B-5C-44.41.10
Comp-Add Latch	C1	3C-4C-44.41.15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Carry/No-Carry Insert		
No-Carry Insert	C1 AP	3C-4C-44. 41. 02
Carry Insert	C1 AP	3F-4G-44. 41. 02
No-Carry 2nd	C1 A - C	4E-5D-44. 41. 01
Carry 2nd	C1 A - C	4J-5H-44. 41. 01
Set Normal Num Sign		
Sign Chg Ctrl	C0-1	4B-44. 35. 65
Sign Insert (+)	C0-1	3E-44. 35. 64
Set Alpha Sign		
Alpha Chg Latch	C0	3B-5C-44. 35. 65
Sign Chg Ctrl	PC Last +	5E-44. 35. 65
Alpha Insert	PC Last +	6E-44. 35. 65

G. Arithmetic Register Read-Out to Adder

The arithmetic register which contains the data address factor enters the adder at Entry B. The register is scanned out with the field ring, which is under control of the instruction field register setting.

During the normal serial read-out for the regeneration cycle, the bit values of all positions are read out to a scan matrix. The matrix sense amplifiers are gated with the field ring pulses to select successive digits. A single position is fed to the output latches for each digit time.

The output latches are gated to the Channel 2 Mix, digit by digit, to feed Adder Entry B. The normal Channel 2 zeros, used continually to validate the channel, are suppressed with the use of true or complement add. The true and complement add signals also gate the adder entry for the proper function.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Scan Out		
Scan Matrix	F 1/2 Time	XX-44. 01. 11
Output Latches (0 Bit)	AP	5E-44. 01. 16

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
RO Athr to Channel 2		
RO Athr to Ch 2 Latch	C1 AP	4B-5B-44.05.42
Channel 2 Mix (0 Bit)	BP	5F-6E-44.46.01
Blank Ch 2 Zeros	C1	4C-5D-42.62.15
Adder Entry B	ASP-B	4X-6X-44.40.06

H. Auxiliary Register RO to Adder

The auxiliary register contains the operated accumulator value. It is serially transferred to Adder Entry A over Channel 1 starting with the low order. The three sections of the auxiliary register operate as a single ten-position register. The serial read-out of one section enters the next section with the low order reading out through the Auxiliary 3 output latch. A single shift gate is developed, which is mixed in the serial controls for each section. Zeros are inserted at the high-order position of the register.

The Auxiliary 3 output latches are gated to the Channel 1 Mix to feed Adder Entry A. The normal Channel 1 zeros used to maintain validity are suppressed by the rise of either true or complement add. The carry or no-carry signals gate the entry to the adder, providing either direct decimal translation or a plus one value.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Start Entire Auxr Adv		
	C0-1	4D-42.31.10
Auxr 1 Ser Adv	C1 AP	5C-5D-42.31.10
Auxr 2 Ser Adv	C1 AP	4E-5D-42.31.11
Auxr 3 Ser Adv	C1 AP	4B-5B-42.31.12
Aux Reg Ser RO		
Auxr 3 Output Latches (0 Bit)	AP	3G-4G-42.31.09
Zero Insert	CP	3F-5F-42.31.47
RO Auxr 3 to Channel 1		
RO Auxr 3 to Ch 1 Latch	C1	4D-5A-42.31.13
Ch 1 Mix (0 Bit)	BP	5E-6E-44.45.01

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Blank Ch 1 Zeros	C1	3E-4E-42.62.10
Adder Entry A	ASP-A	4X-6X-44.40.04

I. Arithmetic Register RI from Adder Output

The sum or difference values of the adder entries is read out to the adder output first latches. The normal zero insert for adder output channel validation is suppressed by the turn-on of the adder RO latch. The latch serves as a delayed turn-on and turn off of the true and complement add signals. A total of one full digit delay occurs between the setting of the adder and the return of the factor to the register.

With an add to accumulator operation, the parallel zero insert latches for the arithmetic register are used to fill unused high-order positions. The latches are all set at the start of the cycle. They are then reset one at a time as the program ring advances. At the end of the adder operation, the latches, which are still set, cause zeros to be inserted in the high-order positions.

The adder output is scanned into the arithmetic register by program ring gating starting with the low-order position. The entry pulse is fed to a read-in winding of each position. The position of entry is selected by the individual read-in drivers.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder RO to AO Channel		
Set RO Adder Latch	C1 CP	4A-5B-44.41.05
Blank AO Zero Insert	C1 CP	5D-44.41.05
Blank AO Clear	CP	4E-44.61.03
Set AO First Latches (0 Bit)	CP	4A-5A-44.40.15
AO Channel Drive (0 Bit)	AP	5D-44.40.10
Set Athr Zero Insert Latches		
Set Zero Insert Latches (P9)	CX	3A-4A-44.04.01
Reset Digit Positions (P9)	C1-2DP	2B-3B-44.04.01
RI Athr Reg from AO		
Athr Add to A Ctrl	C1-2	3A-4B-5E-44.05.35
Athr Add to A Ser RI	AP	3F-4F-5J-44.05.35

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Blank Athr Ser RO (P9)	C1 - 2	3F-4E-5E-44.05.40
Athr Scan RI (P9)	C2 AP	5A-44.05.30
Athr Zero Insert		
Athr Zero Insert Ctrl	PC Last +	3B-44.04.03
Blank Ser RO for ZI (P9)	PC Last +	2E-4E-5E-44.05.40
Athr Insert Ctrl	AP	2A-44.04.03
Zero RI	PC Last + AP	4A-44.04.04

J. Field End Controls

The end-of-adder operation occurs when the end of both factors has been reached. The arithmetic field end is indicated by the field ring match signal. The accumulator field end is indicated by the shift register A last signal. The coincidence of these signals starts the program cycle ring to stop the operation. A true-add operation starts the ring at last minus, while a complement-add starts at last plus. The additional digit on true-add allows processing a high-order carry. A special condition occurs when the arithmetic field ends first. Field control indicates no more usable digits. The register transfer is stopped and zeros are inserted on Channel 2 for the rest of the operation.

The field-ring operation is checked following the match indication by switching the field length register bit latches with the the program ring value to reset the field ring check latch.

Example:

1. A data word is field defined 36. This involves a 4 digit scan-out of the arithmetic register and the last digit scans out at F3 field-ring time which will coincide with C4 program ring time for this example. The 6 which is in the units position of the field register is 9's complement added to the 3 which is in the tens position to give an adder output of 6 (6 and 0 bit). Interrogation of Logic 44.02.22 will show the 6 and 0 bit Field Length Register outputs switching with C4.
2. A data word is field defined 27. The adder output is a value of 4 (3 and 1 bit). This involves a six digit scan out of the arithmetic register. Field Length Register Latch outputs switch with C6 in this case.

The Field Ring Check Latch is turned on at the beginning of each operation and must be turned off to prevent a Field Ring Error indication.

To prove that the significant digit scanner made no error on its original scan, a test is made on the remaining contents of the auxiliary register. The register is read out in parallel to the arithmetic bus and into the scanner. The normal bus clear circuit dumps the remainder of the capacitor charge in the absence of a core read-in. A read-out from the significant digit scanner zero position resets the SDS check latch. Failure to reset the latch signals an error.

A true-add operation in which either factor has the full ten digits, tests adder carry on the tenth digit. A carry at this point indicates that an eleventh digit has been developed. When this occurs, the accumulator overflow latch is set. The latch may be tested for branching control or cause machine stop.

On a complement-add operation the adder no-carry first latch is tested on the last transfer to determine if a carry has developed. The lack of a carry indicates the sign of the total has reversed, and the value in the arithmetic register is in complement form. With this condition, a complement-adjust cycle is initiated to convert the value to true notation and adjust its sign.

The program ring is stopped either by the program cycle test pulse for a normal operation or with a program-ring recycle developed by the complement adjust.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field Ring Match		
Field Stop Match (P3)	F3	3C-44.02.01
Field Ring Last	BP	3A-4A-44.02.02
Field Ring Last Plus	CP	3D-4D-44.02.02
Field Ring Test	AP	3G-4G-44.02.02
Field Ring Stop	CP	3B-4B-44.02.03
Field Ring Error Test		
Field Ring Error Check	F Last	3H-44.02.23
Field Ring Check	CP	4X-5D-44.02.22
Check Latch Reset	CP	4C-5C-44.02.24
Arith Field End		
Arith Fld End Latch	F Last	4B-5B-42.40.25
Ch 2 ZI Latch	AP	5H-5F-42.62.15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Shift Register End		
A Last Latch	AP	4B-5B-44. 50. 06
A Last Plus	CP	4G-5G-44. 50. 06
A Field End	AP	3F-4E-5E-42. 40. 25
Start Program Cycle Ctrl's		
Set Prg Cy Last Minus	CP	3A-4A-5A-42. 40. 22
Prg Cy Last	AP	4D-5D-42. 40. 22
Prg Cy Last Plus	CP	4F-5G-42. 40. 22
Set Prg Cy Last Plus	CP	3G-4G-5G-42. 40. 22
Prg Cy Test	AP	4A-5A-42. 40. 21
Prg Cy Stop	CP	4D-5D-42. 40. 21
Prg Cy Stop Plus	AP	4G-5G-42. 40. 21
Significant Digit Scanner Check		
SDS Recycle	A Last	3B-4B-44. 06. 06
RO Auxr to AB	PC Last +	4E-42. 31. 66
SDS On	CP	4B-44. 06. 09
SDS Check Reset	PC Stop	4E-4H-5H-44. 06. 05
Test for Accumulator Overflow		
True-Add Delayed	C1	6G-44. 41. 10
Overflow Test	C11 AP	2A-44. 42. 01
A1 Overflow Latch	C11 AP	4A-5A-44. 42. 01
Test for Complement Adjust		
Set Comp Adj Latch	PC Last +	3A-4A-44. 41. 20
Program Ring Stop Ctrl		
Normal Stop Inhibit	PC Last +	2J-5H-41. 12. 04
Prg Ring Stop Latch	PC Test	3F-4G-5G-41. 12. 02

K. Complement Adjust Cycle Setup

The program ring is immediately recycled to start the complement adjust cycle. The recycle control stops the ring advance from its previous position and restarts it at the beginning.

The arithmetic register must be scanned out starting with its low-order position. This requires the use of the field ring and a forced setting of the field register. The units position is set to nine and the tens position setting is under control of the program ring value at the last register entry. This value is translated to its 11's complement prior to setting the tens position. The field length register and the field ring are set in the same manner as for the start of the Add operation.

The complement-add control is forced with the complement adjust signal. A carry insert is developed for the first entry in the normal manner for a 10's complement.

The complement adjust end latch is set at the start of the adjust cycle. This signal is used to gate the adjust operation, while the complement adjust is reset to prevent forcing a second adjust cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Recycle Program Ring		
Prg Ring Recycle Latch	PC Test AP	3G-4G-5G-41.12.03
Prg Ring Stop	CP	4H-5G-41.12.02
Recycle Start	AP	3B-4A-41.10.01
Set Field Register		
Set Fld Reg Latches	PC Test	4F-42.06.00
Prg Ring Pos Translate	PC Test	XX-44.02.25
Set Fld Reg TP (0 Bit)	PC Test	5H-5F-42.06.01
Set Fld Reg UP (3 Bit)	PC Test	5E-42.06.04
Set Field Size Register		
Field Reg to Adder	CX	3B-44.02.23
RO Fld Reg TP Ch 1	CX	3B-42.07.01
RO Fld Reg UP Ch 2	CX	3D-42.07.01
Complement-Add	CX	5E-44.41.15
No-Carry Insert	CX AP	4C-44.41.02

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Size Reg Reset	CX BP	2H-44.02.20
RI Fld Size Reg	C0 BP	3E-4E-44.02.23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44.02.20
Set Field Ring and Advance		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Ring SPos (F1-9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
On Time Latches (F9)	AP	4C-5C-44.03.01
1/2 Time Latches (F9-8)	CP	4F-5F-44.03.01
Set Adder Controls		
Set Comp-Add	C1	3B-4C-44.41.15
Set Carry Insert	C1 AP	3F-4G-44.41.02
Set Carry 2nd Latch	AP	4J-5H-44.41.01
Set Comp Adj End	C1 AP	3B-4B-44.41.21

L. Complement Adjust Cycle Adder Operation

The arithmetic register is scanned out to its output latches with the field ring. This results in a serial transfer starting with the low-order position. The output is gated to the Channel 2 Mix to feed Adder Entry B. The normal Channel 2 zeros are suppressed with the complement-add signal. The data is gated to the adder entry with the complement-add signal.

Adder Entry A is fed with zeros from Channel 1. The normal channel zeros are suppressed by the complement-add signal, while the complement adjust end sets the insert latch. The carry and no-carry latch signals gate the zeros to the adder entry.

The adder output is read into the arithmetic register with the program ring in the same manner as the normal add operation. The zero-insert latches are set and reset in the entry positions. The remaining latches cause zeros to be inserted in the high-order positions at the end of the cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr RO to Adder		
Athr Scan Out	F 1/2 Time	XX-44.01.11
Output Latches (0 Bit)	AP	5E-44.01.16
RO Athr to Ch 2	C1 AP	3B-5B-44.05.42
Channel 2 Mix	BP	5F-6E-44.46.01
Blank Ch 2 Zeros	C1	4C-5D-42.62.15
Adder Entry B	ASP-B	4X-6X-44.40.06
Channel 1 Zero Insert		
Blank Normal Ch 1 Zero	C1	3E-4E-42.62.10
Ch 1 ZI Latch	C1	6D-42.62.10
Adder Entry A	ASP-A	4X-6X-44.40.03
Athr RI from Adder		
Adder RO to AO Channel		*
Set Athr Zero Insert		*
RI Athr from AO		*
Athr Zero Insert		*

* See Normal Add Operation (pages 74 and 75)

M. Complement Adjust Field-End Controls

The field-end controls for a complement-adjust cycle are similiar to the add cycle. The A field-end latch must be forced because no accumulator factor is involved. The field ring continues to advance until the match point is reached. With a match the arithmetic field-end latch is set to indicate transfer completion. The normal complement-add condition starts the program cycle at last plus. The program ring is stopped by setting its stop latch at program cycle test.

At the end of the complement adjust cycle, the sign of the arithmetic register is again set. Unless one of the factor signs is alpha, the new sign is set opposite that of the operated-accumulator factor.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A Field End (Forced)	C1	4F-5E-42. 40. 25
Arith Fld End		
Field Ring Match		*
Fld Ring Err Test		*
Arith Fld End		*
Start Prg Cy Ctrls		
Prg Cy Last +	CP	3G-4G-5G-42. 40. 22
Prg Cy Ring		*
Program Ring Stop	PC Test CP	3F-4G-5G-41. 12. 02
Set Arith Sign		
Sign Chg Ctrl	PC Last +	4E-44. 35. 65
Sign Insert Bits (+)	PC Last +	5B-44. 35. 64

*See Normal Add Operation (pages 76 and 77)

N. Arithmetic Register Transfer to Operated Accumulator

When the operation reaches program cycle stop with no complement adjust, the arithmetic register with the new total is transferred to the operated accumulator. The condition may occur either with or without an adjust cycle. The previous accumulator value is blanked before read-in. The arithmetic bus clear or dump control must be blocked during the transfer to allow core read-in.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Par RO to AB	PC Stop	4A-44. 05. 49
Op'd A RI from AB		
Op'd A RI AB Ctrl (A1)	PC Stop	2C-44. 11. 00
A1 RI AB (Block Ser RO)	PC Stop	4A-44. 11. 03
A1 AB RI Latch	AP	3A-4A-44. 11. 01
AB Par Clear Ctrl	PC Stop AP	5E-42. 31. 21

O. End Data Operation

The signal to end the add to accumulator operation occurs just prior to the final transfer at program cycle last plus. The transfer overlaps the request for the next instruction read-in.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Data Op	PC Last +	3A-42. 40. 40
Set EDO Latch	PC Last +	4C-42. 40. 30

Typical Timing Chart

The following timing charts (Figures 5.1-4 and 5.1-5) illustrate the general timings for the add-to-accumulator operation. A specific case is assumed with a ± 14 operation code. The signs produce a complement add operation and result in a complement adjust cycle. One of the factor signs is alpha to illustrate alpha insert and no complement adjust insert. The field control of 36 and an accumulator factor of six significant digits as assumed cannot actually produce the adjust cycle. A data address of Accumulator 2 is used to illustrate this operation. The start of an operation using a core storage address is identical to that for the add to storage operation (Figures 5.1-8 and 5.1-9). The remainder of the operation is the same for any address.

The timing chart is started with a successful data request of memory. The request must be made regardless of the data address. The data latch is turned on earlier during either the instruction or index cycle. Several signals remain on beyond the end data op signal. In most cases they fall as the result of the instruction control signals. The break points shown on these lines are for the earliest memory access. The failure to honor an instruction request delays the fall by twelve microseconds.

The interpret operation signal is representative of the many signals in the operation detail of Section A.

5.1.02 Zero and Add-Subt Accumulator (± 13 , ± 23 , ± 33)

Zero and Add-Subt Absolute Accumulator 1 (± 16)

All of the above codes follow the basic zero and add-to-accumulator operation. The codes differ in the consideration of the signs. This operation differs from a normal add-to-accumulator in that the factor in the operated accumulator is not used. Several short cuts are taken in the process. The operation and the signs are forced to allow the minimum amount of processing. If the field control indicates a full ten-digit word, the adder operation is suppressed and a straight parallel transfer occurs. The following major objectives are required to perform the operation:

- A. Data Cycle Start
- B. Data Read-In to Arithmetic Register
- C. Operated Accumulator Factor Inhibit

- D. Set Field Control
- E. Set Sign and True-Add Latches
- F. Full Field Transfer Controls
- G. Arithmetic Register Read-Out to Adder
- H. Channel 1 Zero Insert
- I. Arithmetic Register Read-In from Adder Output
- J. Field End Controls
- K. Arithmetic Register to Operated Accumulator
- L. End Data Operation

A. Data Cycle Start

The data cycle start for a zero and add-to-accumulator follows the normal arithmetic pattern. When the data latch is set, the word at the specified data address is read into the arithmetic register. The address may be any core storage location or one of the three accumulators.

The interpret operation signal switches with the op-register latch output to determine the zero and add-to-accumulator code. These signals are mixed with those of other codes. Many of the basic add controls are used and the operations inhibited with the special signals.

The normal memory request develops the A & P address and information gate. The address gate starts the data control ring to read in the factors. The program ring is started, in turn, to control the cycle operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Data Latch	IX End	4D-42.40.04
Set Data Req Latch	MOP	5G-42.40.04
A & P Address Gate	MOP-XP	4E-55.01.02
A & P Info Gate	ZP	5D-55.01.04
Start Data Ctrl Ring	Addr Gate	5B-42.41.10
Set CAB Latch Reg	DC1 AP	4G-5F-42.05.01
999X Address	DC1	3H-42.05.05
Core Address	DC1	5D-44.05.50
Interpret Operation	DC3B	3C-4B-42.40.08
Op Reset Add to A	DC3B	5F-42.11.05
Op Add Subt	DC3B	3F-4F-42.11.05
Op RAAB or RSAB	DC3B	3D-42.11.05

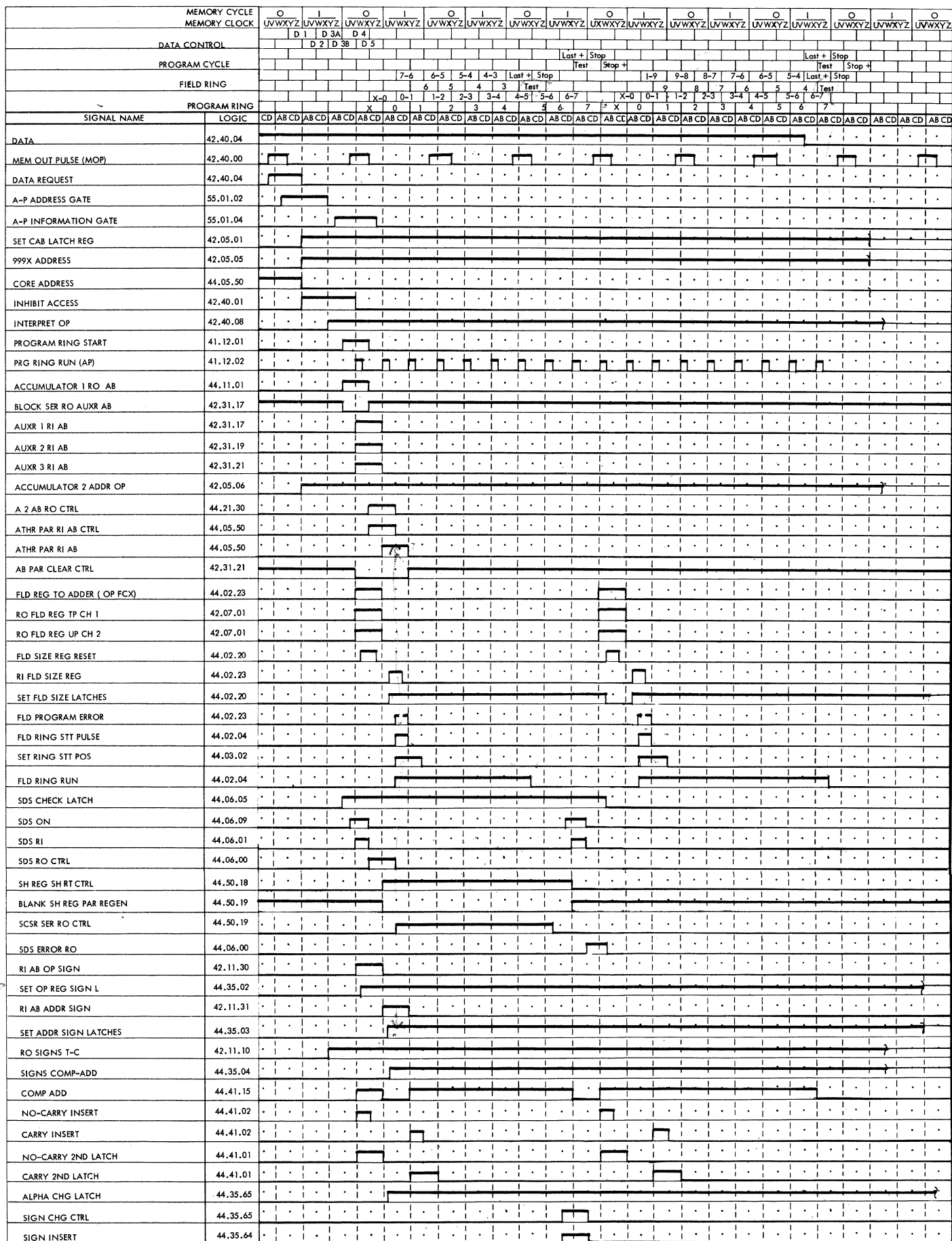


FIGURE 5.1-4. ADD TO ACCUMULATOR

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Op RAAB	DC3B	4C-42.11.05
Op RSAB	DC3B	3C-42.11.05
Op Add to A	DC3B	2F-3F-4F-42.11.05
Op RS	DC3B	4H-42.11.05
Arith Codes	DC3B	3E-42.11.06
Op Matrix True-Add	DC3B	5C-42.11.09
Fld Ctrl Code Mix	DC3B	5A-42.11.16
Op A 1	DC3B	2A-42.11.50
Start Program Ring	DC4 AP	3A-4A-41.10.01
Prg Ring Start	DC4	4D-41.12.01
Prg Ring Run	DC4 AP	5B-41.12.02

B. Data Read-In to Arithmetic Register

The data word may originate in either core storage or one of the accumulators. Separate logic paths are followed for the two areas. The selected word reads into the arithmetic register during the information gate at Data Control 5 core-storage with a core storage address. With an accumulator address the core-storage read-out and the information bus validity checks are inhibited. The selected accumulator transfers to the arithmetic register over the arithmetic bus at C0 time.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Address		
Athr Par RI IB Ctrl	DC4	2E-44.05.51
Blank Ser RO (P0)	DC4	5A-44.05.38
Athr Par RI IB	DC4 AP	4E-5E-44.05.51
999X Address		
Inhibit Access	DC2	4E-5E-42.40.01
Addr A RO to AB		

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A (2) Addr Op	DC3B	4D-5E-42.05.06
A (2) AB RO Ctrl	DC4	3J-44.21.30
Athr Par RI AB Ctrl	CX-0	4C-44.05.50
Blank Ser RO (P0)	CX-0	5A-44.05.38
Athr Par RI AB	C0 AP	5E-5F-44.05.50

C. Operated Accumulator Factor Inhibit

The operated accumulator factor is not required for the zero and add operation. The close timing between the operation code development and the factor timing make it impractical to inhibit the normal read-out of the factor. It is read out to the arithmetic bus and into the auxiliary register. During the transfer the significant digit scanner is set with the location of the high-order digit. The significant digit scanner check latch controls the use of both of these registers. By inhibiting the set of this latch with the op reset add to A signal, the factor is ignored. The scanner read-out controls do not turn on to allow setting the single-core shift register. The factor cannot read out to Channel 1, because the auxiliary register serial advance is not developed.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
SDS Stt Inhibit	DC3B	44.02.26
Inhibit SDS Check Latch		3G-44.06.05
Inhibit SDS RO Ctrl		4B-44.06.00
Inhibit Stt Entire Auxr Adv		4D-42.31.10

D. Set Field Control

The field register is set with the instruction word. The digit values indicate the position and size of the factor within the data word. The field ring is started at the value of the units digit and runs until the match point is reached as indicated by the tens position. The field ring is used to scan the field from the arithmetic register unless a full ten-digit field is specified.

A check is made on the field ring advance to insure that only the specified digits are used. The field length register is set with the 10's complement of the difference between start and stop values. The size is tested against the program ring count value, when the field stop match occurs.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Reg to Adder (Op FCX)	CX	3B-44.02.23
RO Fld Reg TP Ch 1	CX	3B-42.07.01
RO Fld Reg UP Ch 2	CX	3D-42.07.01
Complement-Add	CX	5E-44.41.15
No Carry Ins	CX AP	4C-44.41.02
RI Fld Size from AO		
Fld Size Reg Reset	CX BP	2H-3H-44.02.20
RI Fld Size Reg	C0 BP	3E-4G-44.02.23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44.02.20
Fld Program Error	C0 CP	3E-4E-44.02.23
Set Field Ring		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Ring Pos (F 1 - 9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
Advance Fld Ring		
On Time Latch (F 9)	AP	4C-5C-44.03.01
1/2 Time Latch (F 9 - 8)	CP	4F-5F-44.03.01

E. Sign and True-Add Latches

Sign handling for the zero and add codes differs from the normal add operations because of the omission of one factor. Because a complement adjust cycle is only needed on a complement-add function, this operation is eliminated in favor of forced sign change. The zero and add operation is always performed as a true-add by forcing with the operation code signals. The operated register sign latches are allowed to set during the factor transfer, because they are not required for other signs.

The setting of the addressed sign latches differs between normal and absolute operations. For normal operations, the signs are set from the addressed factor in the arithmetic register. With absolute operations the sign latches are forced to read-in the operation register sign. An add operation sets the plus latch and a subtract operation sets the minus latch.

The sign, except for the absolute zero and add code, is selected from the addressed sign latches and inserted into the arithmetic register sign position. In the zero and add code the arithmetic register contains the correct sign as it came from core storage. For the zero and subtract code, the signs are reversed unless it was alpha originally. Each of the absolute codes have signals developed to select the latch which agrees with the operation sign.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Addr Sign Latches		
Normal Codes		
RI AB Addr Sign	C0	4C-42. 11. 31
Athr RO Sign to AB	CX-0	3B-42. 11. 31
Set Addr Sign Latch (+)	C0	5A-6A-44. 35. 03
Absolute Codes		
Normal Addr Sign Inh	DC3B	3D-2C-42. 11. 31
Op Register Sign Ins	CX	4E-42. 11. 31
Set Addr Sign Latch (+)	CX	4A-6A-44. 35. 03
Set True-Add Latch	C1	3D-5C-44. 41. 10
Set No-Carry		
No-Carry Insert	C1 AP	3C-4C-44. 41. 02
No-Carry 2nd Latch	C1 A - C	4E-5D-44. 41. 01
Set RS Signs		
Sign Chg Ctrl	PC LAST +	4G-44. 35. 65
Plus Insert	C0	3G-5E-44. 35. 64
Minus Insert	C0	3G-4F-44. 35. 64
Alpha Insert	C0	3G-5H-44. 35. 64
Set RAAB and RSAB Signs		
Sign Chg Ctrl	PC LAST +	4J-44. 35. 65
Plus Insert	CX	5C-44. 35. 64
Minus Insert	CX	4G-44. 35. 64

F. Full Field Transfer Controls

When the setting of the field register indicates that the full ten digits of the data word are to be used, the adder operation is eliminated. The entire word is parallel-transferred to the operated accumulator. The detection of the full-field transfer entails inhibiting the start of the field ring and stopping the program ring, which has already been started. The field ring error checks must be satisfied to prevent an error indication.

The normal adder and transfer circuits are inhibited by switching with a Field 1 to 9 signal, which is the "not" full field or field size 10 signal. The program cycle ring is started at the Last position at C1 time to end the operation. The final transfer is identical to the smaller factor and is detailed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Full Field Interpretation		
Fld Size 10	IC4	2B-44.02.26
Fld 1 to 9	IC4	5B-44.02.26
Op Full Fld Tran	DC3B	4F-44.02.26
Field Ring Control		
Fld Stt Inh	DC3B	4F-4G-44.02.26
Fld Chk Latch Reset	C1	4F-4C-44.02.24
Fld Ring Error	C2	5F-44.02.24
Set Prg Cy Last	C1	3G-4G-42.40.20

G. Arithmetic Register Read-Out to Adder

The arithmetic register is serially scanned out to the adder under field ring control. The field ring is started at the specified digit for low order, moving toward the specified high order digit by digit. The ring digit gates the digits from the register to its output latches.

The arithmetic register output latches are read out and gated to the Channel 2 mix to feed Adder Entry B. The normal Channel 2 zeros are suppressed with the rise of the true-add signal. True-add also gates the entry to the adder.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Arithmetic Scan Out		
Scan Matrix	F 1/2 TIME	XX-44.01.11
Output Latches (0 Bit)	AP	5E-44.01.16
RO Athr to Ch 2		
RO Athr to Ch 2 Latch	C1 AP	4B-5B-44.05.42
Channel 2 mix (0 Bit)	BP	5F-6E-44.46.01
Blank Ch 2 Zeros	C1	5C-5D-42.62.15
Adder Entry B	ASP-B	4X-6X-44.40.06

H. Channel 1 Zero Insert

The factor from the operated accumulator is not used in the zero and add operation. The rise of the true-add signal suppresses the normal validity checking of zeros on Channel 1. Zeros are inserted on the channel with the zero insert latch to satisfy both the validity check and the adder entry.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Channel 1 Zero Control		
Blank Ch 1 Zeros	C1	4E-42.62.10
Ch 1 Zero Insert	C1	3C-4B-42.62.10
Adder Entry A	ASP-A	4A-6A-44.40.03

I. Arithmetic Register Read-In from Adder Output

The adder output is scanned back into the arithmetic register, under program ring control, starting with the low-order register position. The arithmetic register zero insert latches are set at the start of the operation and reset, in order, as digits are entered from the adder. At the end of the operation those latches, which are not reset, force zeros into the high-order positions.

The read-out adder controls the setting of the adder output latches and the suppression of the adder output validity check zeros. The output is delayed from the input by one digit time. The normal adder output clear or dump line is suppressed during the transfer.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder RO to AO Channel		
RO Adder Latch	C1 CP	4A-5B-44. 41. 05
Blank AO Zero Insert	C1 CP	5D-44. 41. 05
Blank AO Clear	CP	4E-44. 61. 03
Set AO First Latches (0 Bit)	CP	4A-5A-44. 40. 15
AO Channel Drive (0 Bit)	AP	5D-44. 40. 10
Set Athr Zero Insert Latches		
Set Zero Insert Latches (P9)	CX	3A-4A-44. 04. 01
Reset Digit Positions (P9)	C 1 - 2 DP	2B-3B-44. 04. 01
RI Arith Reg from AO		
Athr Add to A Ctrl	C1-2	3A-4B-5E-44. 05. 35
Athr Add to A Ser RI	AP	3F-4F-5J-44. 05. 35
Blank Athr Ser RO (P9)	C1-2	3F-4E-5E- 44. 05. 40
Athr Scan RI (P9)	C2	5A-44. 05. 30
Athr Zero Insert		
Athr Zero Insert Ctrl	PC LAST +	3B-44. 04. 03
Athr Insert Ctrl	PC LAST +	2A-44. 04. 03

J. Field End Controls

The end of the adder operation occurs when the end of the arithmetic field is reached. The A field-end latch is forced because the accumulator factor is not used. The program cycle ring is started at the last-plus position to stop the operation.

When the field ring stop match is reached, the operation is checked by switching the field length register with the program ring value. A correct operation resets the field ring check latch to indicate no error. The match condition of the ring sets the arithmetic field end latch.

The program ring is stopped by the program cycle test pulse setting the program ring stop latch. This in turn drops the run latch to stop the advance.

A full-field transfer re-enters the problem at this point. The program cycle ring has been forced to complete the operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field Ring Match		
Fld Stop Match (P3)	F3	3C-44.02.01
Fld Ring Last	BP	3A-4A-44.02.02
Fld Ring Last Plus	CP	3D-4D-44.02.02
Fld Ring Test	AP	3G-4G-44.02.02
Fld Ring Stop	CP	3B-4B-44.02.03
Field Ring Error Test		
Fld Ring Error Chk	F LAST	3H-44.03.23
Fld Ring Chk	CP	4X-5D-44.02.22
Check Latch Reset	CP	4C-5C-44.02.24
Athr Fld End Latch	F LAST	4B-5B-42.40.25
A Fld End Latch	C1	3D-5E-42.40.25
Start Program Cycle Ctrls		
Set Prg Cy Last (Full Field)	C1	3G-42.40.20
Set Prg Cy Last (Fld 1 to 9)	AP	
Prg Cy Last Plus	CP	4F-5G-42.40.22
Prg Cy Test	AP	4A-5A-42.40.21
Prg Cy Stop	CP	4D-5D-42.40.21
Prg Cy Stop Plus	AP	4G-5G-42.40.21
Prg Ring Stop Latch	PC TEST	3F-4G-5G-41.12.02

K. Arithmetic Register to Operated Accumulator

When the program cycle ring reaches the stop position, the factor in the arithmetic register is transferred to the operated accumulator. The operation is the same for full field or partial field. The previous accumulator value is blanked before read-in. The arithmetic bus clear or dump control is blocked during the transfer to allow core read-in.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Par RO to AB	PC STOP	4A-44.05.49
Op'd A RI from AB		
Op'd A RI AB Ctrl (A 1)	PC STOP	2C-44.11.00
A 1 RI AB (Block Ser RO)	PC STOP	4A-44.11.03
A 1 AB RI Latch	AP	3A-4A-44.11.01
AB Par Clear Ctrl	PC STOP AP	5E-42.31.21

L. End Data Operation

The signal to end a zero and add to accumulator occurs, just prior to the final factor transfer, at program-cycle last-plus. The transfer overlaps the request for the next instruction read-in.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Data Op	PC LAST +	3A-42.40.40
Set EDO Latch	PC LAST +	4C-42.40.30

Typical Timing Chart

The following timing chart (Figures 5.1-6 and 5.1-7), illustrates the general timings for a zero and add operation. A specific case is assumed with a - 13 operation code. The factor sign is of no consequence in the operation. The field control is assumed to be 04 for the basic chart with notations for 09 or full-field transfer. The use of a core-storage address is shown. The use of an accumulator address would change the start of the operation to that similar to that for add to accumulator (Figures 5.1-4 and 5.1-5).

The timing chart is started with a successful data request of core storage. The data latch was turned on earlier during either the instruction or index cycle. Several signals remain on beyond the end-data-op signal. In most cases, they fall as the result of instruction control signals. The break points shown on these lines are for the earliest

memory access. The failure to honor instruction request delays the fall by twelve microseconds.

The interpret operation signal is representative of all of the operation signals shown in Section A.

- 5.1.03 Add to storage from Accumulator# (+18, +28, +38)
 Subtract Accumulator # from storage (-18, -28, -38)
 Add to Absolute Storage from Accumulator# (+19, +29, +39)

All of the above codes follow a basic add-to-storage operation with variations in the sign control. Much of the operation resembles the add-to-accumulator operation (Section 5.1.01) except for the differences in gating signals and in the reversal of factors. The following major objectives are required to perform the operation:

- A. Data Cycle Start
- B. Data Read-In to Arithmetic Register
- C. Operated Accumulator Transfer to Auxiliary Register
- D. Set Field Control
- E. Operated Accumulator Factor Size Control
- F. Set Sign and True-Complement Latches
- G. Arithmetic Register Read-Out to Adder
- H. Auxiliary Register Read-Out to Adder
- I. Arithmetic Register Read-In from Adder Output
- J. Field End Controls
- K. Complement Adjust Cycle Setup
- L. Complement Adjust Cycle Adder Operation
- M. Complement Adjust Field End Controls
- N. Store Arithmetic Register
- O. End Data Operation

Figures 5.1-8 and 5.1-9 show the sequence of operation.

A. Data Cycle Start

The data-cycle start for add-to-storage follows the normal arithmetic pattern. When the data latch is set, the word at the specified data address is read into the arithmetic register. The data address may be any core-storage location or one of the three accumulators.

The interpret operation signal switches with the op-register latch output to determine the add-to-memory code. This signal is mixed with those of other codes to form several signals used throughout the operation.

The data request of memory, when accepted, develops the A & P address and information gates. The address gate starts the data control ring to read in the factors. The program ring is started in turn to control the cycle operation.

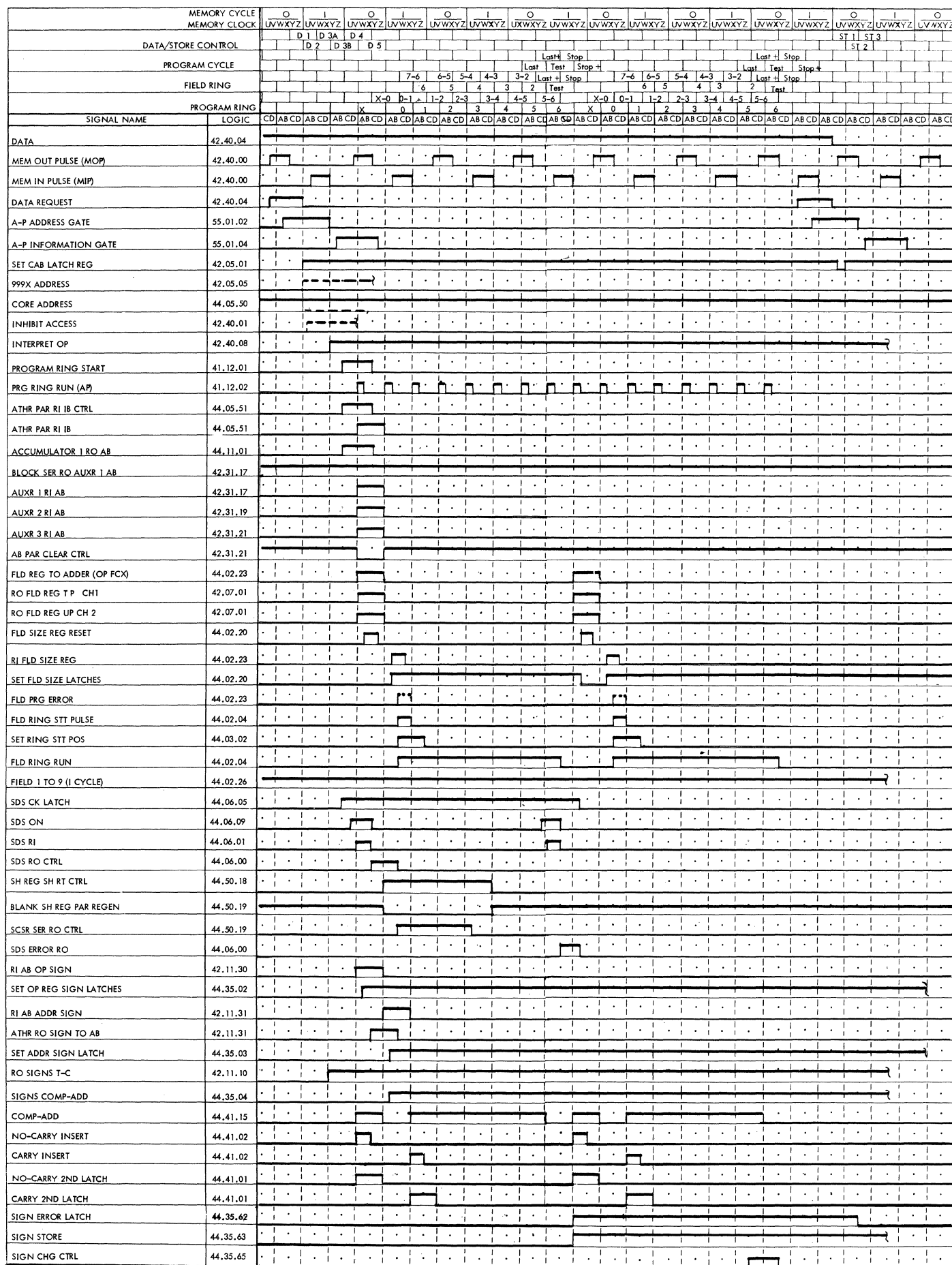


FIGURE 5.1-8. ADD TO STORAGE

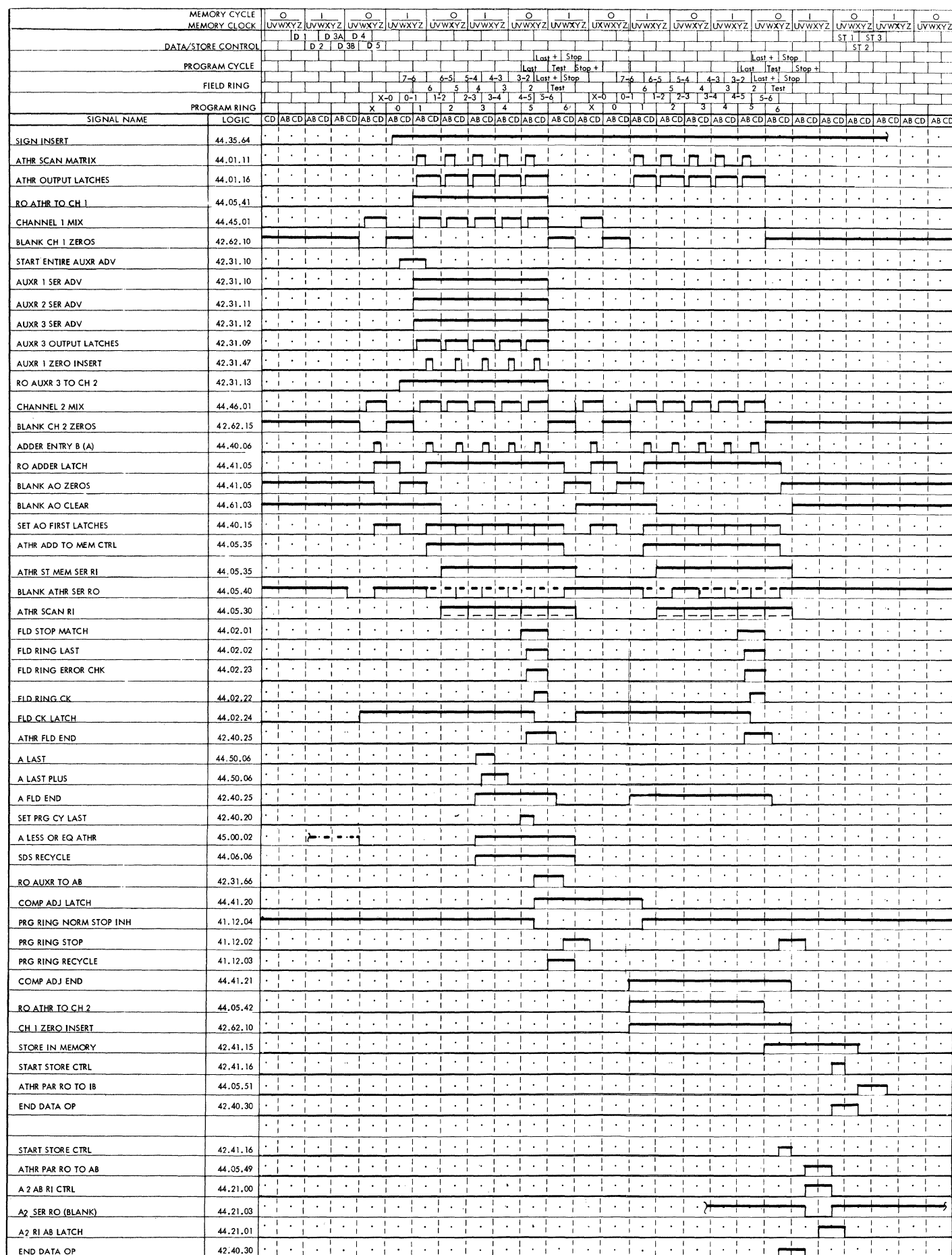


FIGURE 5.1-9. ADD TO STORAGE

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Data Latch	IX End	4D-42.40.04
Set Data Req Latch	MOP	5G-42.40.04
A & P Address Gate	MOP-XP	4E-55.01.02
A & P Info Gate	ZP	5D-55.01.04
Start Data Ctrl Ring	Addr Gate	5B-42.41.10
Set CAB Latch Reg	DC1 AP	4G-5F-42.05.01
999X Address	DC1	3H-42.05.05
Core Address	DC1	5D-44.05.50
Interpret Operation	DC3B	3C-4B-42.40.08
Op Add-Subt to Mem	DC3B	2C-42.11.15
Add to Mem Codes	DC3B	3C-42.11.15
Op Add-Subt	DC3B	4A-42.11.05
Store in Mem Codes	DC3B	4B-42.11.04
Arith Codes	DC3B	3E-42.11.06
Fld Ctrl Code	DC3B	5A-42.11.16
Op A1	DC3B	2A-42.11.50
Start Program Ring	DC4 AP	3A-4A-41.10.01
Prg Ring Start	DC4	4D-41.12.01
Prg Ring Run	DC4 AP	5B-41.12.02

B. Data Read-In to Arithmetic Register

The data word may originate in either core storage or one of the accumulators. Separate logic paths are followed for the two areas. The address is sent to the core area during the A & P address gate. The selected word reads into the arithmetic register during the A & P information gate at Data Control 5.

An accumulator address causes the core storage read-out and the information bus validity checks to be inhibited. The selected accumulator transfers to the arithmetic register over the arithmetic bus at C0 time.

The data word factor, and thus the arithmetic register, is in effect an accumulator. The operated accumulator factor, therefore, appears as the operand.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Address		
ATHR Par RI IB Ctrl	DC4	2E-44.05.51
Blank Ser RO (P0)	DC4	5A-44.05.38
ATHR Par RI IB	AP	4E-5E-44.05.51
999X Address		
Inhibit Access	DC2	4E-5E-42.40.01
Addr A RO to AB		
A (2) Addr Op	DC3B	4D-5E-42.05.06
A (2) AB RO Ctrl	DC4	3J-44.21.30
ATHR Par RI AB Ctrl	CX-0	4C-44.05.50
Blank Ser RO (PO)	CX-0	5A-44.05.38
ATHR Par RI AB	C0 AP	5E-5F-44.05.50

C. Operated Accumulator Transfer to Auxiliary Register

The operation code specifies the accumulator by the digit value of its ten's digit. The accumulator transfers to the auxiliary register via the arithmetic bus and is re-generated in the normal manner.

The auxiliary register for this operation functions as a ten-position register. Each section is switched independently for the read-in and the normal serial read-out for regeneration is suppressed.

The arithmetic bus capacitor clear circuits are blocked during the transfer because of the core read-in. The significant digit scanner also reads in during the transfer with capacitor sense amplifiers. The scanner operation is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Accumulator 1 RO AB	DC4	2F-44.11.01
Block Ser RO Auxr 1 AB (2 & 3)	DC4	3C-42.31.17

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Auxr RI from AB		
Auxr 1 RI AB	DC4 AP	5A-6A-42. 31. 17
Auxr 2 RI AB	DC4 AP	4D-5D-42. 31. 19
Auxr 3 RI AB	DC4 AP	4B-5B-42. 31. 21
AB Par Clear Ctrl	DC4 A-A	5H-42. 31. 21

D. Set Field Control

The field register is read in from the instruction to denote the size of the field within the data word. In this operation the effective accumulator size is equal to the specified field. The units position of the register is used to start the field ring at the indicated low-order digit of the field. The ring progresses digit by digit until its position matches the ten's position of the field register. The field ring is used to scan the factor out and the total into the arithmetic register. Only the specified positions are used and changed.

A check is made on the field ring advance to insure that the remainder of the word is left intact. The field length register is set to the ten's complement of the difference between the start and stop values. The size is tested against the program ring value, when the match occurs. The error check is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field 1 to 9	I Cycle	2B-5B-44. 02. 26
Fld Reg to Adder (OP FCX)	CX	3B-44. 02. 23
RO Fld Reg TP Ch 1	CX	3B-42. 07. 01
RO Fld Reg Up Ch 2	CX	3D-42. 07. 01
Complement-Add	CX	5E-44. 41. 15
No-Carry Ins	CX AP	4C-44. 41. 02
RI Fld Size from AO		
Fld Size Reg Reset	CX BP	2H-3H-44. 02. 20
RI Fld Size Reg	C0 BP	3E-4G-44. 02. 23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44. 02. 20
Fld Program Error	C0 CP	3E-4E-44. 02. 23

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Fld Ring		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Ring Pos. (F 1 - 9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
Advance Field Ring		
On Time Latch (F 9)	AP	4C-5C-44.03.01
1/2 Time Latch (F 9 - 8)	CP	4F-5F-44.03.01

E. Operated Accumulator Factor Size Control

The significant digit scanner register is set with the location of the high-order digit location during the operated accumulator transfer. The indication is immediately transferred to the single-core shift register by reading directly into the serial capacitors. The shift register right shift control is used for read-in. The shift register is used as a ring to count the serial transfer of the factor to the adder. When the indicator bit serially reads out of the low-order position, all of the significant digits of the factor have been transferred. The field-end detection is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set SDS Check Latch	DC3 A	3G-4G-5G-44.06.05
SDS On	DC4 DP	3A-44.06.09
SDS RI (9P)	DC4 AP	3D-44.06.01
SDS RO to SCSR		
SDS RO Ctrl	CX-0 AP	4B-44.06.00
Sh Reg Sh Rt Ctrl	CX-0 AP	2B-4C-5B-44.50.18
Blank Sh Reg Par Regen	CX-0 AP	3B-44.50.19
SCSR Ser RO Ctrl	CX-0 CP	4F-44.50.19

F. Set Sign and True-Complement Latches

With a normal add-subtract to storage operation the signs of both factors and the sign of the operation are matrixed to determine the actual function to be performed. The sign of the operated accumulator is sensed during the time when it is transferred on the arithmetic bus. The addressed factor sign is read-in one digit time later. If the address is an accumulator, it is read in during the transfer. When the factor comes from a core storage address, the sign alone is read out from the arithmetic register to the arithmetic bus for read-in. In setting the true-complement latches, an alpha sign is considered plus. The factor signs are compared for like and unlike and then switched with the operation sign to determine the function.

The add to absolute storage from accumulator # operation differs from the basic add to storage in the sign consideration. The storage sign is never changed. The sign of the addressed factor is ignored and is treated as an absolute value. The sign of the operated accumulator may be either plus or minus making the factor either an increment or a decrement. The addressed sign latches are set plus, while the operated accumulator sign is set in the normal manner. If a complement adjust signal is developed, the correction cycle is taken but the sign change is not made.

The sign latches remain set until the end of the operation to allow either an alpha change or a sign reversal for complement adjust to be made or indicated. The arithmetic register sign is not changed at the start of the operation as with the add to accumulator operation. The arithmetic register factor is the effective accumulator in this case. An alpha operated accumulator sign causes a sign change if the addressed sign is other than alpha.

When complement adding, an adder carry insert is developed for the first digit entry. The true-add signal develops a no-carry insert. On subsequent digit times the adder develops its own carry and no-carry signals.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Op'd Reg Sign Latches		
RI AB Op Sign	CX	3B-5A-42. 11. 30
Set Op Reg Sign Latch (+)	CX	5A-6A-44. 35.02
Set Addr Sign Latches		
Normal Add		
RI AB Addr Sign	C0	4C-42. 11. 31
ATHR RO Sign to AB	CX-0	3B-42. 11. 31
Set Addr Sign Latch (+)	C0	5A-6A-44. 35.03
Add Absolute		
Normal Addr Sign Inh	DC3B	3E-2C-42. 11. 31

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Op Reg Sign Ins	CX	4E-42, 11. 31
Set Addr Sign Latch +	CX	4A-6A-44, 35. 03
RO Signs T - C	DC3B	3B-42, 11. 10
Signs True-Add+ (++)	C0	3B-5B-44, 35. 04
Signs Comp-Add+ (+-)	C0	3E-3F-44, 35. 04
Set T-C Latches		
True-Add Latch	C1	3B-5C-44, 41. 10
Comp-Add Latch	C1	3C-4C-44, 41. 15
Set Carry/No-Carry Insert		
No-Carry Insert	C1 AP	3C-4C-44, 41. 02
Carry Insert	C1 AP	3F-4G-44, 41. 02
No-Carry 2nd Latch	C1 A-C	4E-5D-44, 41. 01
Carry 2nd Latch	C1 A-C	4J-5H-44, 41. 01
Set Alpha Sign		
Set Mem Alpha	C1	4C-44, 35. 62
Sign Error Latch	C1	6C-44, 35. 62
Alpha Chg Latch	C1	5C-44, 35. 65
Sign Chg Ctrl	PC Last +	5E-44, 35. 65
Alpha Insert	PC Last +	6E-44, 35. 65

G. Arithmetic Register Read-Out to Adder

The arithmetic register with the effective accumulator factor from the data address enters the adder at Entry A. The register digits are scanned out during the normal regeneration read-out by the field ring. The ring is under control of the field register set from the instruction word. The serial output from the register starts from the low order specified by the field register and continues digit by digit to the specified high order.

The arithmetic register output latches are read out and gated to the Channel 1 Mix to feed Adder Entry A. The normal Channel 1 zeros are suppressed with the rise of true or complement add. The carry or no-carry signals gate the entry to the adder.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Arithmetic Scan Out		
Scan Matrix	F 1/2 Time	XX-44.01.11
Output Latches (0 Bit)	AP	5E-44.01.16
RO ATHR to Ch 1		
RO ATHR to Ch 1 Latch	C1 AP	4B-5B-44.05.41
Channel 1 Mix (0 Bit)	AP	5F-6E-44.45.01
Blank Ch 1 Zeros	C1	3E-4E-42.62.10
Adder Entry A	ASP-A	4X-6X-44.40.03

H. Auxiliary Register RO to Adder

The auxiliary register contains the factor from the operated accumulator, which for the add to storage operation is the operand. The register is serially transferred to Adder Entry B starting with the low order. The three sections of the auxiliary register function as a single ten-position register using the Auxiliary 3 output latches. A single-shift control is developed which is mixed in the serial controls for each section. Zeros are inserted at the high order during the transfer.

The Auxiliary 3 output latches are gated to the Channel 2 Mix to feed Adder Entry B. The normal channel 2 validity zeros are suppressed with the rise of either true or complement add. The true-add and complement-add signals also gate the entry to the adder.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Start Entire AUXR Adv.	C0-1	4D-42.31.10
AUXR 1 Ser Adv.	C1 AP	5C-5D-42.31.10
AUXR 2 Ser Adv.	C1 AP	4E-5D-42.31.11
AUXR 3 Ser Adv.	C1 AP	4B-5B-42.31.12
AUXR Ser RO		
AUXR 3 Output Latches (0Bit)	AP	3G-4G-42.31.09

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Zero Insert	CP	3F-5F-42. 31. 47
RO AUXR 3 to Channel 2		
RO AUXR 3 to Ch 2 Latch	C1	4G-4E-42. 31. 13
Ch 2 Mix (0 Bit)	AP	5E-6E-44. 46. 01
Blank Ch 2 Zeros	C1	5C-5D-42. 62. 15
Adder Entry B	ASP-B	4X-6X-44. 40. 06

I. Arithmetic Register RI from Adder Output

The totals resulting from the adder entries are read out to the adder-output-first latches. The normal adder-output channel zeros are suppressed by the turn-on of the adder RO latch. There is a delay of one digit time between the arithmetic register read-out and the result read-in.

With an add to storage operation, the parallel zero insert latches of the arithmetic register are not used. The latches are still set and reset but do not fall in a usable position. The remainder of the word in the register remains unchanged.

The adder output is scanned into the arithmetic register by field ring gating starting with the same low order position that was read out one digit earlier. The field ring selects the proper read-in driver.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder RO to AO Channel		
RO Adder Latch	C1 CP	4A-5B-44. 41. 05
Blank AO Zero Insert	C1 CP	5D-44. 41. 05
Blank AO Clear	CP	4E-44. 61. 03
Set AO First Latches (0 Bit)	CP	4A-5A-44. 40. 15
AO Channel Drive (0 Bit)	CP	5D-44. 40. 10
RI Arith Reg from AO		
ATHR Add to Mem Ctrl	C1 - 2	3A-4B-5D-44. 05. 35
ATHR Stt Mem Ser RI	AP	3F-4F-5H-44. 05. 35

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Blank Arith Ser RO (P9)	F 9 - 8	2F-4E-5E-44.05.40
ATHR Scan RI (P9)	F8 AP	3A-44.05.30

J. Field End Controls

The end of the adder operation occurs when the end of the arithmetic field is reached. Further adder operation, even if the accumulator field is larger, is of no use because there is no space to enter the total. The program cycle ring is started at last position to stop the operation immediately.

The field-ring operation is checked following the match indication by switching the field-length register with the program ring value to reset the field ring check latch. The match condition sets the arithmetic field-end latch.

The significant digit scanner is recycled to check for possible digits left after the transfer, when the operated accumulator factor is smaller than the arithmetic factor. The auxiliary register is read out to the arithmetic bus and the significant digit scanner. A zero factor output resets the check latches to indicate no error. When the arithmetic factor is the smaller, the set of the arithmetic field end before accumulator field end causes a field overflow indication. The scanner recycle is suppressed, and the check latch reset is forced to prevent an error indication.

A true-add operation, which produces a carry output on the last pair of digits, sets the field overflow indication. A complement-add operation, which fails to produce a carry on the last pair of digits, sets the complement-adjust latch to indicate the value in the arithmetic field is in complement notation. With this condition, a complement-adjust cycle is initiated to correct the condition. A sign change is indicated if the field is less than ten digits.

The program ring is stopped either by the program-cycle test pulse for a normal operation or a program-ring recycle developed by the complement adjust.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field Ring Match		
Field Stop Match (P3)	F3	3C-44.02.01
Field Ring Last	BP	3A-4A-44.02.02
Field Ring Last Plus	CP	3D-4D-44.02.02
Field Ring Test	AP	3G-4G-44.02.02
Field Ring Stop	CP	3B-4B-44.02.03

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field Ring Error Test		
Field Ring Error Check	F Last	3H-44.02.23
Field Ring Check	CP	4X-5D-44.02.22
Check Latch Reset	CP	4C-5C-44.02.24
ATHR Field End Latch	F Last	4B-5B-42.40.25
Shift Register End		
A Last Latch	AP	4B-5B-44.50.06
A Last Plus	CP	4G-5G-44.50.06
A Field End	AP	3F-4E-5E-42.40.25
Start Program Cycle Ctrls		
Set Prg Cy Last	CP	4H-42.40.20
Prg Cy Last Latch	CP	5D-42.40.22
Prg Cy Last Plus	CP	4F-5G-42.40.22
Prg Cy Test	AP	4A-5A-42.40.21
Prg Cy Stop	CP	4D-5D-42.40.21
Prg Cy Stop Plus	AP	4G-5G-42.40.21
Significant Digit Scanner Check		
ATHR Eq or Larger		
A Less or Eq ATHR A Last	AP	3D-5F-45.00.02
SDS Recycle	AP	3A-4B-44.06.06
RO AUXR to AB	PC Last	4E-42.31.66
SDS On	CP	4B-44.06.09
SDS Check Reset	PC Stop	4E-4H-5H-44.06.05
ATHR Smaller		
ATHR Smaller than A	AP	3B-5B-45.00.02
SDS Check Reset	PC Stop	4F-44.06.06

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Test for Field Overflow		
A Field Size	F Last+ CP	3D-4D-4B-44.42.04
Carry Of1	F Last+	2B-4B-44.42.04
Test for Complement Adjust		
Set Comp Adj Latch	PC Last+	3A-4A-44.41.20
Program Ring Stop Ctrl		
Normal Stop Inhibit	PC Last+	2J-5H-41.12.04
Prg Ring Stop Latch	PC Test	3F-4G-5G-41.12.02

K. Complement Adjust Cycle Setup

The program ring is recycled to start the complement adjust cycle. The arithmetic register is scanned out with the field ring. The ring is set from the field register with the same information that was used to perform the add operation. The field register is again read out to the adder and the difference read into the field size register.

The complement-add control is forced with the complement adjust signal. A carry insert is developed for the first entry in the normal manner for a tens complement.

The complement-adjust end latch is set at the start of the adjust cycle. This signal is used to gate the adjust operations, while the complement-adjust latch is reset to prevent forcing a second adjust cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Recycle Program Ring		
Prg Ring Recycle Latch	PC Test AP	3G-4G-5G-41.12.03
Prg Ring Stop	CP	4H-5G-41.12.02
Recycle Start	AP	3B-4A-41.10.01
Set Fld Size Register		
Fld Reg to Adder	CX	3B-44.02.23
RO Fld Reg TP Ch 1	CX	3B-42.07.01

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
RO Fld Reg UP Ch 2	CX	3D-42.07.01
Complement-Add	CX	5E-44.41.15
No-Carry Insert	CX AP	4C-44.41.02
Fld Size Reg Reset	CX BP	2H-44.02.20
RI Fld Size Reg	C0 BP	3E-4E-44.02.23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44.02.20
Set Field Ring and Advance		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Fld Ring Pos (F 1 - 9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
On Time Latches (F 9)	AP	4C-5C-44.03.01
1/2 Time Latches (F 9 - 8)	CP	4F-5F-44.03.01
Set Adder Controls		
Set Comp-Add	C1	3B-4C-44.41.15
Set Carry Insert	C1 AP	3F-4G-44.41.02
Set Carry 2nd Latch	AP	4J-5H-44.41.01
Set Comp Adj. End	C1 AP	3B-4B-44.41.21

L. Complement Adjust Cycle Adder Operation

The arithmetic register is scanned out to its output latches with the field ring. This results in a serial transfer, starting with the low order, of the same field used during the add cycle. The output is gated to the Channel 2 Mix to feed Adder Entry B. The complement-add signal suppresses the normal channel 2 zeros and gates the adder entry.

Adder Entry A is fed with zeros from Channel 1. The normal zeros are suppressed by complement-add, while the insert latch is set by complement adjust end. The carry insert and subsequent developed carry signals gate the adder entry.

The adder output is read into the same area of the arithmetic register as just read out with the field ring. The operation is the same as for the earlier add cycle. The zero insert latches for the arithmetic register are not used because the remaining portions of the word must remain unchanged.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
ATHR RO to Adder		
Athr Scan Out	F 1/2 Time	XX-44.01.11
Output Latches (0 Bit)	AP	5E-44.01.16
RO Athr to Ch 2	C1 AP	3B-5B-44.05.42
Channel 2 Mix	AP	5F-6E-44.46.01
Blank Ch 2 Zeros	C1	4C-5D-42.62.15
Adder Entry B	ASP-B	4X-6X-44.40.06
Channel 1 Zero Insert		
Blank Normal Ch 1 Zeros	C1	3E-4E-42.62.10
Ch 1 ZI Latch	C1	6D-42.62.10
Adder Entry A	ASP-A	4X-6X-44.40.03
ATHR RI from Adder		
Adder RO to AO Channel		*
RI ATHR from AO		*

*See Normal Add Operation (page 74)

M. Complement Adjust Field End Control

The field end controls for a complement-adjust cycle are similar to the add cycle. The A field-end latch is forced by complement adjust because no accumulator factor is involved. When the field ring match point is reached, the arithmetic field-end latch is set to indicate the end of the transfer. The program cycle ring is started by setting the last position of the ring. The program ring is stopped by setting its stop latch at program cycle test.

At the end of the complement-adjust cycle, the sign of the arithmetic register is reversed if it was numeric. If the sign of either factor was alpha, the result sign is alpha after the adjust cycle. The sign of the core storage word having been changed sets the sign change indication. The sign is not changed on an add to absolute storage from accumulator # operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A Fld End (Forced)	C1	4F-5E-42. 40. 25
Fld Ring Match		*
Fld Ring Error Test		*
ATHR Fld End		*
Start Prg Cy Ctrls		
Set Prg Cy Last	CP	4H-42. 40. 20
Prg Cy Ring		*
Program Ring Stop	PC Test CP	3F-4G-5G-41. 12. 02
Set ATHR Sign		
Sign Error	CX	5D-6C-44. 35. 62
Sign Store	CX	3C-4C-44. 35. 63
Sign Chg Ctrl	PC Last+	5F-44. 35. 65
Sign Insert Bits	PC Last	5E-44. 35. 64

*See Normal Add Operation (pages 76 and 77).

N. Store Arithmetic Register

The result of the operation in the arithmetic register must be returned to the data address location. Because the address may be either core storage or one of the accumulators, there are two possible logic paths. The previously set computer address bus latches indicate which path is to be used.

A core storage address sets the store in memory latch at program cycle test following either the normal add cycle or the adjust cycle. At memory-in pulse time, the store request latch is set. The return of an address gate from core storage starts the store control ring. Store Control 1 reads out the data address to core storage. At store control 3 the arithmetic register is read out in parallel to the information bus. The result enters the core storage location replacing the original data.

An accumulator address short cuts the operation by forcing the store control ring. The arithmetic register is read out in parallel to the arithmetic bus at Store Control 3 time. The same signal sets the read-in for the specified accumulator replacing the original data.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Storage Address		
Store in Memory	PC Test	4F-5F-42. 41. 15
Store Request	MIP	2B-3B-42. 41. 14
Start Store Ctrl Ring	Addr Gate CP	4C-5C-42. 41. 16
ATHR Par RO to IB	STC3	3A-5B-44. 05. 51
999X Address		
Start Store Ctrl Ring	PC Test CP	3C-5C-42. 41. 16
ATHR Par RO AB	STC3	4B-44. 05. 49
A2 AB RI Ctrl	STC3	2G-44. 21. 00
A2 RI AB	AP	3D-4D-44. 21. 01

O. End Data Operation

The signal to end the add-to-storage operation occurs just prior to the final transfer. It is initiated by Store Control 1. The transfer overlaps the request for the next instruction. If the instruction request is granted immediately, the instruction word is transferred on the memory cycle following the data transfer.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Data Op		
Set EDO Latch	STC1	3C-4C-42. 40. 30

Typical Timing Chart

The timing charts (Figures 5.1-8 and 9) illustrate the general timings for the add-to-storage operation. A specific case is assumed with a ± 18 Operation Code. The signs produce a complement-add operation and result in a complement-adjust cycle. A field control of 26 and an accumulator factor of five significant digits are assumed. A data address in core storage is shown in the general portion of the chart. The start of an operation using an accumulator address is identical to that for the add-to-accumulator operation (Figures 5.1-4 and 5). An insertion at the end of the chart shows the finish of an operation with accumulator address.

The timing chart is started with a successful data request of memory. A few signals including the data latch are turned on during the instruction or index cycles.

Several signals remain on beyond the end-data-op signal. In most cases they fall as the result of the instruction control signals. The break points shown on these lines are for the earliest memory access. The failure to honor an instruction request delays the fall by twelve microseconds.

The interpret operation signals are representative of the many signals in the operation detail, Section A.

- 5.1.04 Zero Storage and Store Accumulator # (-11, -21, -31)
- Store Accumulator # (+12, +22, +32)
- Store Digits from Accumulator # and ignore sign, (Store Absolute) (-12, -22, -32)

All of the above codes follow a basic store-in-memory operation. They are similar to an add-to-storage operation except that the field area of the data word is not added. The store codes vary in sign considerations in much the same manner as their add counterparts. If the operated accumulator factor is larger than the specified field, the overflow indicator is set as in the add operations.

Special cases are made when the specified field is for a full ten digits. The entire operated accumulator with its sign is placed in core storage. Overflow errors are not possible, but a sign change is detected for the +12 code for program control. The adder operation is dispensed with on a full field and the factor is parallel transferred. The following major objectives are required to perform the operation:

- A. Data Cycle Start
- B. Data Read-In to Arithmetic Register
- C. Operated Accumulator Transfer to Auxiliary Register
- D. Set Field Control
- E. Operated Accumulator Factor Size Control
- F. Set Signs and True-Add Latches
- G. Full Field Transfer Controls
- H. Auxiliary Register Read-Out to Adder
- I. Channel 1 Zero Insert
- J. Arithmetic Read-In from Adder Output
- K. Field End Controls
- L. Store Arithmetic Register
- M. End Data Operation

Figures 5.1-10 and 11 show the sequence of operations.

- A. Data Cycle Start

The start of the operation for a store code is identical to that for an add-to-storage or other arithmetic operation. The analysis for full or partial field is made after the rise of the interpret operation signal. When the data latch is set, the word at the specified data address is read into the arithmetic register. The address may be any core storage location or one of the accumulators. The operated accumulator factor is transferred to the auxiliary register.

The interpret operation signal switches with the op register latch output to determine the store code. These signals are mixed with those of other codes. The movement of data is similar to other codes and special signals are used only for selective operations.

The normal memory request develops the A & P address and information gates. The address gate starts the data control ring to read-in the factors. The program ring is started in turn to control the cycle operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Data Latch	IX End	4D-42.40.04
Set Data Request Latch	MOP	5G-42.40.04
A & P Address Gate	MOP-XP	4E-55.01.02
A & P Info Gate	ZP	5D-55.01.04
Interpret Operation	DC3B	3C-4B-42.40.08
Op Store Codes	DC3B	2C-3A-42.11.04
Store in Mem Codes	DC3B	2C-3A-5B-42.11.04
RSTM Codes	DC3B	2D-3E-42.11.04
Op Store 1-2-3	DC3B	2C-4G-42.11.04
Arith Codes	DC3B	3G-42.11.06
Op Matrix True-Add	DC3B	5B-42.11.09
Fld Ctrl Code	DC3B	5A-42.11.16
Op A 1	DC3B	2A-42.11.50
Start Program Ring	DC4 AP	3A-4A-41.10.01
Prg Ring Start	DC4	4D-41.12.01
Prg Ring Run	DC4 AP	5B-41.12.02

B. Data Read-In to Arithmetic Register

The data word may originate in either core storage or one of the accumulators. Separate logic paths are followed for the two areas. The selected word from core storage reads into the arithmetic register at Data Control 5. With an accumulator the memory read-out is inhibited. The transfer from the selected accumulator occurs on the arithmetic bus at C0 time.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Address		
Athr Par RI IB Ctrl	DC4	2E-44.05.51
Blank Ser RO (P0)	DC4	5A-44.05.38
Athr Par RI IB	C0 AP	4E-5E-44.05.51
999X Address		
Inhibit Access	DC2	4E-5E-42.40.01
Addr A RO to AB		
A (2) Addr Op	DC3B	4D-5E-42.05.06
A (2) AB RO Ctrl	DC4	3J-44.21.30
Athr Par RI AB Ctrl	CX-0	4C-44.05.50
Blank Ser RO (P0)	CX-0	5A-44.05.38
Athr Par RI AB	C0 AP	5E-5F-44.05.50

C. Operated Accumulator Transfer to Auxiliary Register

The operation code specifies the accumulator to be used by its tens digit. The accumulator parallel transfers via the arithmetic bus and regenerates in the normal manner.

The factor reads into the auxiliary register, which for this operation functions as a ten-digit register. Each section is switched independently for read-in and the normal serial read-out for regeneration is suppressed.

The arithmetic bus capacitor clear or dump circuits are suppressed during the transfer to allow setting the cores on read-in. The significant digit scanner also reads in during the transfer with capacitor sense amplifiers. The scanner operation is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Accumulator 1 RO AB	DC4	2F-44.11.01
Block Ser RO Auxr 1 AB (2&3)	DC4	3C-42.31.17

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Auxr RI from AB		
Auxr 1 RI AB	DC4 AP	5A-6A-42. 31. 17
Auxr 2 RI AB	DC4 AP	4D-5D-42. 31. 19
Auxr 3 RI AB	DC4 AP	4B-5B-42. 31. 21
AB Par Clear Ctrl	DC4 A-A	5H-42. 31. 21

D. Set Field Control

The field register is set with the instruction word. The digit values indicate the position and size of the factor within the data word. The field ring is started at the value of the units digit and runs until a match is indicated with the tens position. The field ring is used to scan the accumulator factor into the specified field in the arithmetic register. If a full field transfer is indicated the field ring is not started.

A check is made on the field ring advance to insure that only the specified positions are used. The field length register is set with the tens complement of the difference between the start and stop values. The adder with complement-add control is used to determine the difference. The size is tested against the program ring count value when the field stop match occurs. An error is also indicated if the field register digits are reversed.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Reg to Adder (Op FCX)	CX	3B-44. 02. 23
RO Fld Reg TP Ch 1	CX	3B-42. 07. 01
RO Fld Reg UP Ch 2	CX	3D-42. 07. 01
Complement-Add	CX	5E-44. 41. 15
No-Carry Insert	CX AP	4C-44. 41. 02
RI Fld Size from AO		
Fld Size Reg Reset	CX BP	2H-3H-44. 02. 20
RI Fld Size Reg	C0 BP	3E-4G-44. 02. 23
Set Fld Size Latches (6B)	C0 CP	4A-5A-44. 02. 20
Fld Prg Error	C0 CP	3E-4E-44. 02. 23

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Field Ring		
Fld Ring Stt Pulse	C0 CP	3A-5A-44.02.04
Set Ring Pos (F 1 - 9)	C0 CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
Advance Fld Ring		
On Time Latch (F9)	AP	4C-5C-44.03.01
1/2 Time Latch (F 9 - 8)	CP	4F-5F-44.03.01

E. Operated Accumulator Factor Size Control

The significant digit scanner register is set with the location of the high-order digit location during the operated accumulator transfer. The indication is transferred to the shift register unless a full field transfer is to be made. In this case the indication is lost during the first read-out drive, because the read-out control has been inhibited as discussed later. Entry into the shift register is through its serial read-out capacitors, reading in with a right shift. The shift register is used as a ring to count the serial transfer of this factor to the adder. When the indicator bit serially reads out of the low order position, all of the significant digits of the factor have been transferred. The field-end detection is discussed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
SDS Check Latch (Fld 1 - 9)	DC3 A	3G-4G-5G-44.06.05
SDS On	DC4 DP	3A-44.06.09
SDS RI (9P)	DC4 AP	3D-44.06.01
SDS RO to SCSR		
SDS RO Ctrl	CX-0 AP	4B-44.06.00
Sh Reg Sh Rt Ctrl	CX-0 AP	2B-4C-5B-44.50.18
Blank Sh Reg Par Regen	CX-0 AP	3B-44.50.19
SCSR Ser RO Ctrl	CX-0 CP	4F-44.50.19

F. Set Signs and True-Add Latches

The store codes differ in the consideration of the signs. On a normal store operation (+12) the operated accumulator sign is always inserted and any sign change is indicated as an error. The condition is the same for the zero storage and store accumulator operation (-11) except that the sign change is not indicated. In the store absolute operation (-12) the operated accumulator sign is ignored except on full field, when the sign is used without sign change indication.

The signs of both the addressed word and the operated factor are set in normal manner for an add operation. The operated accumulator sign is set during the transfer to the auxiliary register. The addressed sign is sensed from the arithmetic bus one digit later regardless of the data address location.

The signs are not used to determine the setting of the true-complement latches. The only reason for the adder operation is to transfer the factor for field positioning. For this purpose the true-add latch is forced with the operation code. With true-add a no-carry insert is generated for the first adder digit.

The signs as set in the latches are compared. The various conditions of both high and low are mixed for the unequal sign signal. This signal is used on the normal store to set the sign error indication. The sign latches remain set until the end of the operation to allow inserting a sign change.

The sign for a full word transfer moves with the factor from the operated accumulator to the arithmetic register.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Operated Reg Sign Latches		
RI AB Op Sign	CX	3B-5A-42. 11. 30
Set Op Reg Sign Latches (+)	CX	5A-6A-44. 35. 02
Set Address Sign Latches		
RI AB Addr Sign	C0	4C-42. 11. 31
Athr RO Sign to AB	CX-0	3B-42. 11. 31
Set Addr Sign Latch (+)	C0	5A-6A-44. 35. 03
Set True Add	C1	3D-5C-44. 41. 10
Set No-Carry Insert		
No-Carry Insert	C1 AP	3C-4C-44. 41. 02
No-Carry 2nd	C1 A-C	4E-5D-44. 41. 01

<u>Major Signal</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set RSTM Sign		
Sign Chg Ctrl	PC Last +	4H-44. 35. 65
Sign Insert	PC Last +	5D-44. 35. 64
Normal Store Sign		
Sign Error Latch	C1	5B-6C-44. 35. 62
Sign Error Store	C1	4D
Sign Chg Ctrl	PC Last +	5F-44. 35. 65
Sign Insert	PC Last +	5D-44. 35. 64

G. Full Field Transfer Controls

When the field register indicates that the full ten-digits of the core storage word are to be replaced by the factor in the operated accumulator, the operation is changed. The serial adder operation is eliminated and the word is parallel-transferred with its sign. The detection of the full field entails inhibiting the start of the field ring and the setting and operation of the shift register. The program ring is stopped by starting the program cycle ring at the last-plus position with the C1 pulse.

The normal transfer to the arithmetic and auxiliary registers at the start of the operation are not used. The auxiliary register is read out in parallel to the arithmetic bus and transferred to the arithmetic register. The normal serial adder transfer is inhibited by the lack of a Field 1 to 9 signal. The final store operation is identical to that for the smaller factor as detailed later.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Full Field Interpretation		
Fld Size 10	IC 4	2B-44. 02. 26
Fld 1 to 9	IC 4	5B-44. 02. 26
Op Full Fld Tran	DC3B	3E-4E-44. 02. 26
Store Full Fld	DC3B	3E-44. 02. 26
Fld Ring Control		
Fld Start Inh	DC3B	3E-4G-44. 02. 26

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Check Latch Reset	C1	4F-4C-44.02.24
Fld Ring Error	C2	5F-44.02.24
Set Prg Cy Last		
Auxr Trans to Athr		
Auxr 1 RO AB	C0-1	6H-42.31.66
Athr Par RI AB Ctrl	C0-1	4B-44.05.50
Blank Ser RO (P0)	C0-1	5A-44.05.38
Athr Par RI AB	C1	4B-5E-5F-44.05.50
Blank AB Par Clear	C1	3B-42.22.02
Op'd A Factor Ser Inh		
SDS Stt Inhibit	DC3B	44.02.26
Inhibit SDS Check Latch		3G-44.06.05
Inhibit SDS RO Ctrl		4B-44.06.00
Inh Stt Entire Auxr Adv		4D-42.31.10

H. Auxiliary Register RO to Adder

The auxiliary register contains the operated accumulator value. It is serially transferred to Adder Entry B over Channel 2 starting with the low order. The three sections of the register operate as a single ten-position register. A single shift gate is developed, which is mixed in the serial controls for each section. The factor is read out from the Auxiliary 3 Output Latches. Zeros are inserted at the high order position of the register.

The Auxiliary 3 output latches are gated to the Channel 2 mix to feed Adder Entry B. The normal Channel 2 zeros are suppressed by the rise of true-add. The true-add signal also gates the adder entry providing direct decimal translation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Start Entire Auxr Adv	C0-1	4D-42.31.10
Auxr 1 Ser Adv	C1 AP	5C-5D-42.31.10
Auxr 2 Ser Adv	C1 AP	4E-5D-42.31.11

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Auxr 3 Ser Adv	C1 AP	4B-5B-42. 31. 12
Auxiliary Reg Ser RO		
Auxr 3 Output Latches (0 Bit)	AP	3G-4G-42. 31. 09
Zero Insert	CP	3F-5F-42. 31. 47
RO Auxr 3 to Ch 2		
RO Auxr 3 to Ch 2 Latch	C1	4H-4E-42. 31. 13
Ch 2 Mix (0 Bit)	AP	5E-6E-44. 46. 01
Blank Ch 2 Zeros	C1	4C-5D-42. 62. 15
Adder Entry B	ASP-B	4X-6X-44. 40. 06

I. Channel 1 Zero Insert

The factor in the specified field of the data word in the arithmetic register is not used. The rise of the true-add signal suppresses the normal Channel 1 zeros. To prevent validity errors, zeros must be inserted for the duration of the operation. The zeros are also required for proper adder operation. The adder entry is gated with the no-carry signal.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Channel 1 Zero Control		
Blank Ch 1 Zeros	C1	4E-42. 62. 10
Ch 1 Zero Insert	C1	3B-4B-42. 62. 10
Adder Entry A	ASP-A	4A-6A-44. 40. 03

J. Arithmetic Register Read-In from Adder Output

The adder output is scanned into the specified area of the arithmetic register with the field ring. Entry starts with the low-order position of the field. The remainder of the word is unchanged in the normal store operations (± 12). In the zero storage and store accumulator # operation (-11), the arithmetic register is cleared and zeros inserted in all positions prior to the adder read-in. The arithmetic zero insert latches are used to enter the zeros.

The read-out adder signal controls the setting of the adder output latches and the suppression of the adder output zeros. The output is delayed from its entry by one digit time. The normal adder output clear or dump line is inhibited during the transfer.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder RO to AO Channel		
Set RO Adder Latch	C1 CP	4A-5B-44.41.05
Blank AO Zero Insert	C1 CP	5D-44.41.05
Blank AO Clear	CP	4E-44.61.03
Set AO First Latches (0 Bit)	CP	4A-5A-44.40.15
AO Channel Drive (0 Bit)	AP	5D-44.40.10
Set Athr Zero Insert Latches		
Set Zero Insert Latches (P9)	CX	3A-4A-44.04.01
Athr Zero Insert		
Athr Zero Ins Ctrl	CX-0	2C-44.04.03
Blank Ser RO for ZI (P9)	CX-0	2E-4E-5E-44.05.40
Athr Insert Ctrl	AP	2A-44.04.03
Zero RI	C0 AP	4A-44.04.04
RI Athr from AO		
Athr Add to Mem Ctrl	C1-2	3A-4B-5D-44.05.35
Athr St Mem Ser RI	AP	2F-4E-5E-44.05.40
Blank Athr Ser RO (P9)	F9-8	2F-4E-5E-44.05.40
Athr Scan RI (P9)	F8 AP	3A-44.05.30

K. Field End Controls

The end of the adder operation occurs when the end of the specified field is reached. Further operation, even if the accumulator field is larger, is of no use because there is no space to enter additional digits. The match condition of the field ring sets the arithmetic field-end latch. The program cycle ring is started in the Last position to stop the operation. The field-ring operation is checked following the match indication by switching the field length register with the program ring. A correct operation resets the field ring check latch.

When the operated accumulator factor is smaller than the specified field, the significant digit scanner is recycled to check for possible digits left after the transfer. The auxiliary register is read out to the arithmetic bus and the significant digit scanner. A zero factor output resets the check latch to indicate no error. When the specified field is the smaller, the set of the arithmetic field end before the accumulator field end causes a field overflow indication. The scanner recycle is suppressed and the check latch reset is forced to prevent an error indication.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Field Ring Match		
Fld Stop Match (P3)	F3	3C-44.02.01
Fld Ring Last	BP	3A-4A-44.02.02
Fld Ring Last Plus	CP	3D-4D-44.02.02
Fld Ring Test	AP	3G-4G-44.02.02
Fld Ring Stop	CP	3B-4B-44.02.03
Field Ring Error Test		
Fld Ring Error Chk	F Last	3H-44.02.23
Fld Ring Chk	CP	4X-5D-44.02.22
Check Latch Reset	CP	4C-5C-44.02.24
Athr Fld End Latch	F Last	4B-5B-42.40.25
Shift Register End		
A Last Latch	AP	4B-5B-44.50.06
A Last Plus	CP	4G-5G-44.50.06
A Fld End	AP	3F-4E-5E-42.40.25
Start Program Cycle Ctrls		
Set Prg Cy Last	AP	4H-42.40.20
Prg Cy Last	AP	5D-42.40.22
Prg Cy Last Plus	CP	4F-5G-42.40.22
Prg Cy Stop	AP	4A-5A-42.40.21
Prg Cy Stop	CP	4D-5D-42.40.21

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Prg Cy Stop Plus	AP	4G-5G-42.40.21
Significant Digit Scanner Check		
Athr Eq or Larger		
A Less or Eq Athr	A Last AP	3D-5F-45.00.02
SDS Recycle	AP	3A-4B-44.06.06
RO Auxr to AB	PC Last +	4E-42.31.66
SDS On	CP	4B-44.06.09
SDS Check Reset	PC Stop	4E-4H-5H-44.06.05
Athr Smaller		
Athr Smaller than A	AP	3B-5B-45.00.02
SDS Check Reset	PC Stop	4F-44.06.06
Field Overflow	F Last + CP	3D-4D-4B-44.42.04
Program Ring Stop	PC Test CP	3F-4G-5G-41.12.02

L. Store Arithmetic Register

The resulting word in the arithmetic register must be returned to the data address location. Because the address may be either core storage or one of the accumulators, there are two possible logic paths. The previously set computer address bus latches indicate which path is to be used.

A core storage address sets the store in memory latch at program cycle test. At memory-in pulse time, the data-request latch is set. The return of an address gate from core storage starts the store control ring. Store Control 1 reads out the data address to core storage. At Store Control 3 the arithmetic register is read out in parallel to the information bus. The word enters the specified storage location replacing the original word.

An accumulator address forces the store control ring. The arithmetic register is read out in parallel to the arithmetic bus at Store Control 3 time. The same signal sets the read-in for the specified accumulator replacing the original data.

A store full-field operation differs in that the memory request is forced at C0 for a core storage address or the store control ring is started at C0 for an accumulator address.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Core Storage Address		
Store in Memory	PC Test	4F-5F-42.41.15
Store Request	MIP	2B-3B-42.41.14
Start Store Ctrl Ring	Addr Gate CP	4C-5C-42.41.16
Athr Par RO to IB	STC3	3A-5B-44.05.51
999X Address		
Start Store Ctrl Ring	PC Test CP	3C-5C-42.41.16
Athr Par RO AB	STC3	4B-44.05.49
A2 AB RI Ctrl	STC3	2G-44.21.00
A2 RI AB	AP	3D-4D-44.21.01
Store Full Field		
Core Address		
Store in Memory	C0	3F-5F-42.41.15
999X Address		
Start Store Ctrl Ring	C0	4A-5C-42.41.16

M. End Data Operation

The signal to end the store operations occurs just prior to the final transfer. It is initiated by Store Control 1. The transfer overlaps the request for the next instruction. If the instruction request is granted immediately, the instruction word is transferred on the memory cycle following the data transfer.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Data Op		
Set EDO Latch	STC1	3C-4C-42.40.30

Typical Timing Chart

The following timing charts (Figures 5.1-10, 11, and 12), illustrate the general timings for a store operation. A specific case is assumed with a -11 operation code. A field control of 37 and an accumulator factor of three significant digits is assumed. The conditions of sign change and field overflow are also shown to illustrate their position. A data address in core storage is shown in the main portion of the chart. The start of an operation using an accumulator address is identical to that for the add-to-accumulator operation (Figures 5.1-4 and 5.1-5). The insertion at the end of the chart shows the finish of an operation with accumulator address.

The full-field transfer condition is shown with broken signal lines. Arrows are shown on the basic cycle controls to indicate the portion of the cycle that is skipped. The program cycle and store control indices are no longer in correct relation to each other, but serve to show the turn-off and store timings.

The timing chart is started with a successful data request of memory. A few signals including the data latch are turned on during the instruction or index cycles. Several signals remain on beyond the end-data-op signal. In most cases they fall as the result of the instruction control signals unless they represent a condition latch.

The interpret-operation signal is representative of the many signals in the operation detail, Section A.

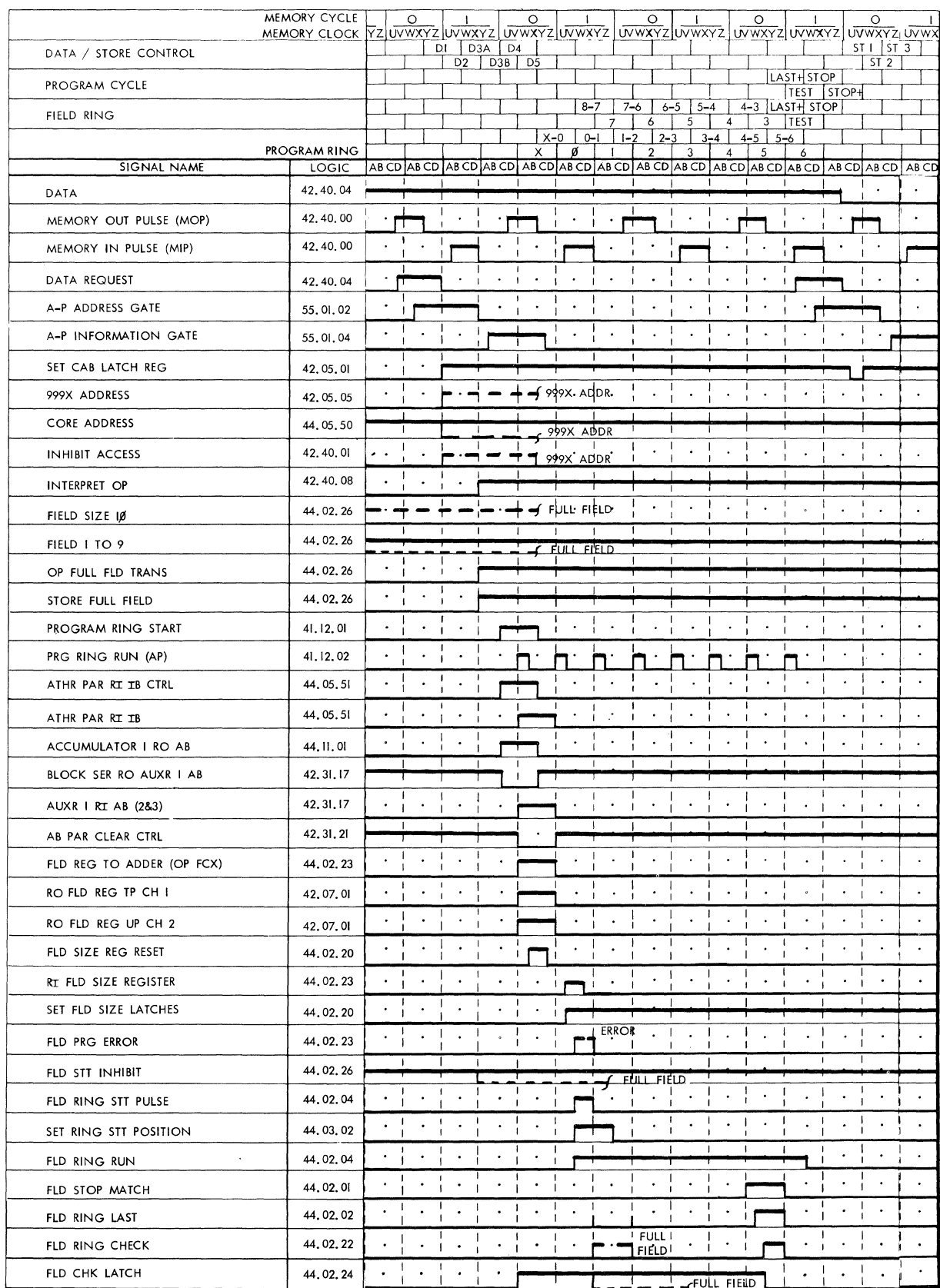


FIGURE 5.1-10. STORE

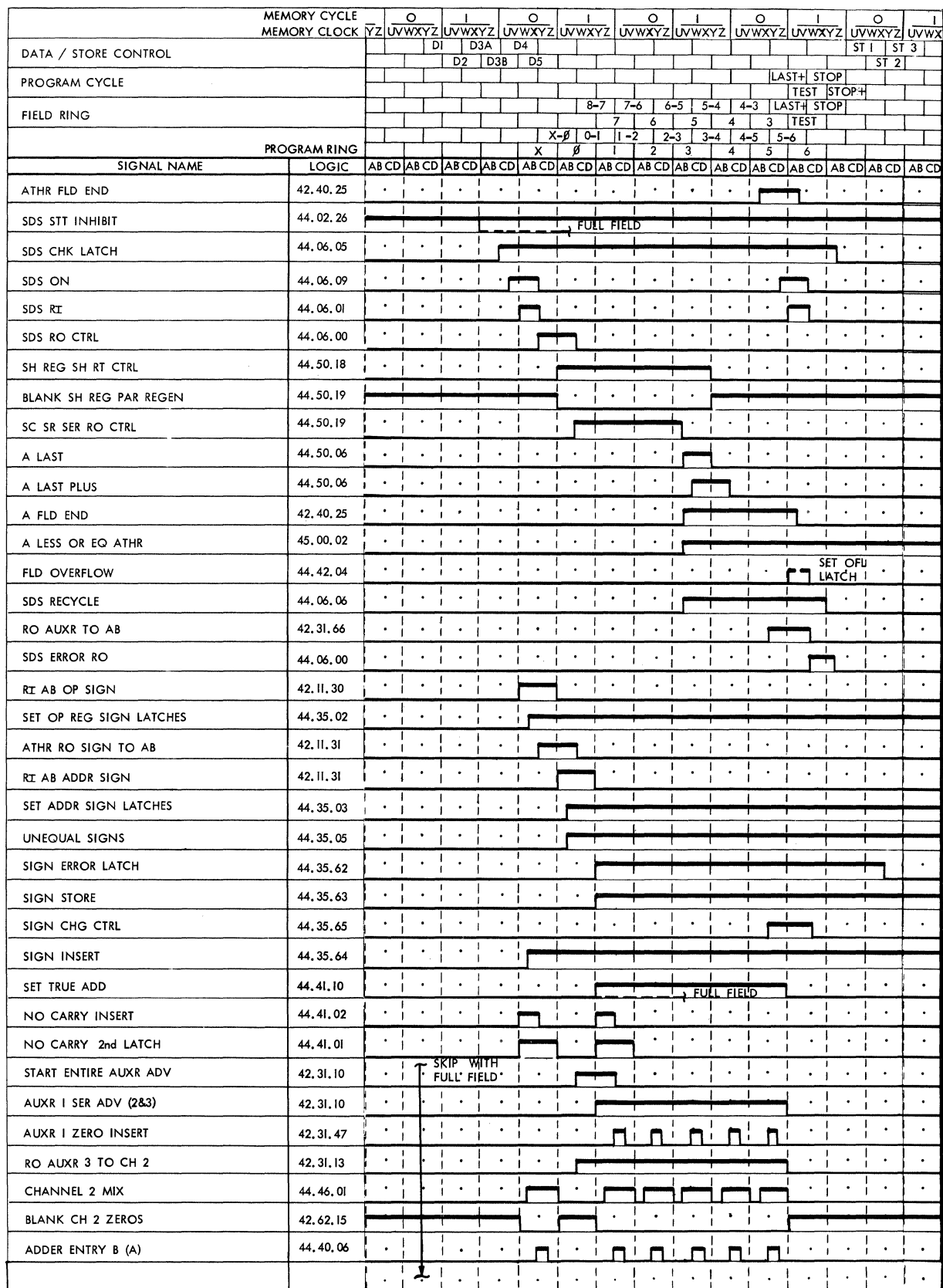


FIGURE 5.1-11. STORE

[illegible]

5.2.00 MULTIPLY (+53) - INTRODUCTION

Multiplication is accomplished by a quadrupler system. The multiplicand factor is first doubled and then quadrupled by addition. The 1X, 2X, and 4X values are carried in three registers to be used in combination to build to the value of the multiplier factor. Thus, a multiplier factor of 2 causes the 2X multiplicand to add, while a multiplier factor of 9 adds the 4X value twice and the 1X value once for a total of three multiply-add cycles. The multiplier digits are processed sequentially starting with the low order. The product accumulator is shifted to the right after each multiplier digit to align the entries. The operation continues until each of the ten multiplier digits has been processed.

The following table shows the multiply-add cycles required for each multiplier factor.

MULTIPLIER FACTOR SELECTION

<u>Multiplier Factor</u>	<u>Multiplicand Factors</u>	<u>Multiply-Add Cycles</u>
0	None	0
1	1X	1
2	2X	1
3	2X + 1X	2
4	4X	1
5	4X + 1X	2
6	4X + 2X	2
7	4X + 2X + 1X	3
8	4X + 4X	2
9	4X + 4X + 1X	3

A multiplicand of up to ten digits is entered into Accumulator 3 prior to the multiply operation instruction. The instruction data address specifies the location of the multiplier factor. The factor enters similar to a zero and add operation. If the field register reads "09" the data word is parallel-transferred directly through the arithmetic register without processing. With a shorter field, the factor is selected and located in the arithmetic register by an adder cycle before transfer to accumulator 2 on the arithmetic bus. A product of up to twenty digits is developed in accumulator 1 and shifted right to fill both accumulators 1 and 2. Accumulator 1 is the high order. The multiplier factor initially entered in Accumulator 2 is lost. The multiplicand factor reappears in accumulator 3 at the end of the operation.

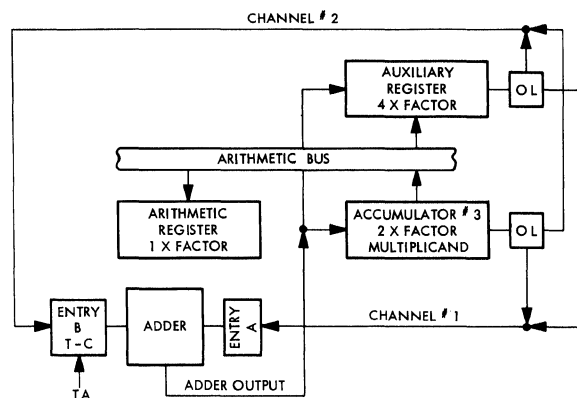


FIGURE 5.2-1. MULTIPLICAND FACTOR DATA FLOW

Multiply Setup

The operation starts by producing the 2X and 4X multiplicand factors. Figure 5.2-1 shows the data flow paths for this portion of the operation. Accumulator 3 is parallel-transferred to the arithmetic bus and regenerated. The data enters the arithmetic register, where it is retained as the 1X factor.

The 2X and 4X factors may, on occasion, contain eleven digits. To accommodate the eleventh digit, the units position is transferred to the output latches, while the remaining ten digits are in the register proper. This system prohibits serial read-out and regeneration of these registers and requires a combination parallel and serial read-out with a second register.

To develop the 2X factor, accumulator 3 is serially read out to both channel 1 and 2. Channel 1 enters adder entry A and channel 2 enters adder entry B with true-add control. The adder output is returned serially to accumulator 3 high order and is shifted until the low order is in the output latches.

The 4X factor is developed by first parallel-transferring accumulator 3 to the arithmetic bus, while the output latches are read to both channels 1 and 2. The data on the arithmetic bus enters the auxiliary register and starts to serially read-out to both channels 1 and 2 on the following digit time. Channel 1 feeds adder entry A, while channel 2 feeds adder entry B with true-add control. The adder output is returned serially to the auxiliary register high order and is shifted until the low order is in the output latches.

Accumulator 1 is set to zeros during the development of the 2X factor by a ten-digit left shift, while zeros are inserted at the low order. During subsequent multiply right shift operations, when the adder is not operating, zeros are inserted into the high order of accumulator 1 from the adder output zero insert.

Factor Selection

Accumulator 2, containing the multiplier factor, is right shifted as the problem progresses to feed the factor digit by digit to the shift register. Each digit is analyzed by transferring the digit to the shift register parallel output latches to test for the use of the 4X, 2X, and 1X factors. After selection, the digit is returned to the shift register and reduced by the value of the selected factor. It is again transferred to the output latches to test for a second factor. This process is repeated until the multiplier

factor digit is zero, then a new digit is read in. Each factor selected causes the corresponding multiplicand factor to be added to the product.

During the first shift of accumulator 2, a five-bit tag digit is read into the high-order position to be used as a signal to stop multiplication, when it reads out of the low-order position. If the first multiplier digit or digits are zeros, the shift is continued with zeros from the low order of accumulator 1 entering the high order of accumulator 2 as the start of the product. Zeros in other positions of the multiplier allow digits of the developed product to transfer.

Multiply - Add

Figure 5.2-2 shows the data flow paths for adding the three multiplicand factors to the product. These factors, as selected, are added to accumulator 1 by successive addition, similar to the over and over addition system.

When the 4X factor is selected, the auxiliary register low-order position in the output latches is read out to channel 1 while the register is parallel-transferred to the arithmetic bus and is regenerated. The data is read into the word buffer register and starts to serial transfer on the following digit time to channel 1 and adder entry A. Simultaneously, accumulator 1 is serially transferred starting with its output latches to channel 2 and adder entry B with true-add control. The adder output is serially returned to accumulator 1 high order and is shifted until its low order is in the output latches.

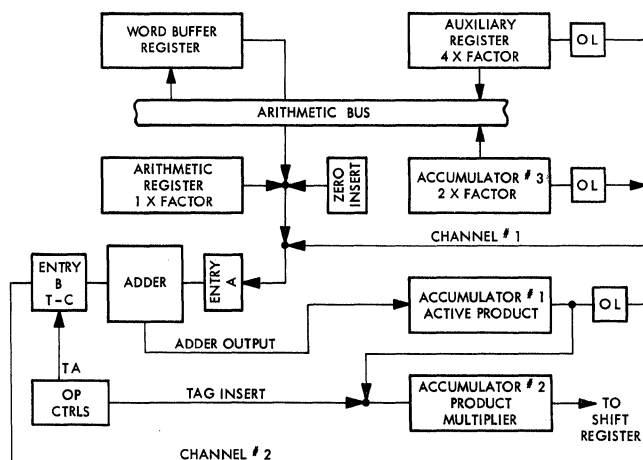


FIGURE 5.2-2. MULTIPLY ADD CYCLE DATA FLOW

The selection of the 2X factor differs only in that accumulator 3 is read out instead of the auxiliary register. The 1X factor, having only ten digits, does not require the second register. The arithmetic register is serially scanned to channel 1 and adder entry A. To match the eleven digits being fed to channel 2, a zero is inserted in the high order from the channel 1 digit insert.

The one or more selected factors are added to the same location in the product. The product is then shifted moving the low-order position from accumulator 1 into accumulator 2. Simultaneously, accumulator 2 low order reads out to the shift register to select the next factor. This process repeats until the tag digit reads out to stop

the operation. The 1X factor in the arithmetic register is then parallel-transferred back to accumulator 3 over the arithmetic bus. Use of the product in accumulators 1 and 2 require additional instructions on subsequent program steps.

Sign

The sign of the product in a multiply operation is normally determined algebraically. If the signs of the multiplier and multiplicand are alike, the product is plus. If the factor signs are different, the product is minus. An exception occurs if either factor sign is alpha, then the product is set alpha.

The sign is determined by the setting of the sign latches on the arithmetic bus. The multiplier sign was set on read-in while the multiplicand sign was set when the factor was moved to the arithmetic register for the 1X factor. At the end of multiply, accumulators 1 and 2 are each parallel-transferred to the arithmetic register and back over the arithmetic bus. While they are in the arithmetic register, the signs are set as required.

5.2.01 + 53 Multiply

The multiply operation is made up of a series of steps, most of which are simplified adding cycles. These steps include reading in the multiplier factor specified by program register D, forming the multiples of the multiplicand factor, and multiply add cycles. These steps are selected and sequenced by a group of multiply operation latches, which provide the necessary operation gates.

Selection of the multiples to be used for a multiplier digit is done by analyzing the parallel output latches of the shift register. As the selected factor is added, the multiplier digit is reduced by shifting and then returned to the output latches for the next selection. If the digit reduces to zero, circuit gates are set to shift the product and the multiplier factor to provide a new digit. Zeros within the factor cause a variation in the cycle to shift the product additional places until a significant digit is reached. A five-bit tag is used in the multiplier register to indicate the end of operation.

The operation is detailed by steps or cycles. A multiplier factor of 106 is used to illustrate a sequence of events. Each multiple is used in this example and the presence of zeros makes use of the multiply idle controls. The timing charts (Figures 5.2-3 to 9) show the same 106 factor with the charts arranged in order. Dotted signal levels show possible signal timings for other factors. The product development for a larger factor is shown in Figure 5.2-10. A simplified timing chart (Figure 5.2-11) shows the cycle progression. The details for the individual cycles can be seen in the previous charts.

The following major cycles are used to perform a multiply operation (Each is detailed in the following):

- A. Zero and Add to Accumulator 2
- B. Form Times Two Multiple
- C. Form Times Four Multiple
- D. Multiply Add - X4 Multiple
- E. Multiply Add - X2 Multiple
- F. Multiply Idle - Zero Factor Digits
- G. Multiply Add - X1 Multiple
- H. End Multiply - Sign Adjust

A. Zero and Add to Accumulator 2 (Figure 5.2-3)

The multiplier factor comes from the location specified by positions 6-9 of the instruction. It enters accumulator 2 in exactly the same manner as a zero and add operation. Field control is used and either the full word or any portion may be specified. When less than the full word is used, the factor is moved to the low-order positions. Data previously in accumulator 2 is lost.

The first cycle is set up as soon as the multiply op code is developed. The setting of the multiply reset add to A latch provides the signal to force the op A2, add to A, reset add to A, and op matrix true-add signals needed for a zero and add. The remaining signals and gates are the same as for the basic operation described in Section 5.1.02.

The first step in the sign control is performed as part of the basic zero and add. The sign of the multiplier factor enters the address sign latches where it remains until the end of the operation. As in zero and add, the operated accumulator sign is set in the op'd A sign latches. The latter sign is not used and is later reset. The sign entering accumulator 2 is that of the word specified by program register D. This sign remains until the final cycle, when it is adjusted to the sign of the product.

The program ring is allowed to run normal in this cycle with a program cycle turn-off. The program cycle stop pulse is used to force a special start for the next step. The effect is the same as a recycle control with CX immediately following the last pulse of the first cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Op Multiply	DC3B	4A-42.11.13
Mpy Reset Add to A	DC3B	3A-4A-45.50.01
Op A2	DC3B	3E-42.11.50
Op Reset Add to A	DC3B	42.11.05
Op Add to A	DC3B	42.11.05
Op Matrix True Add	DC3B	42.11.09
Basic Reset Add Signals	See Section 5.1.02	

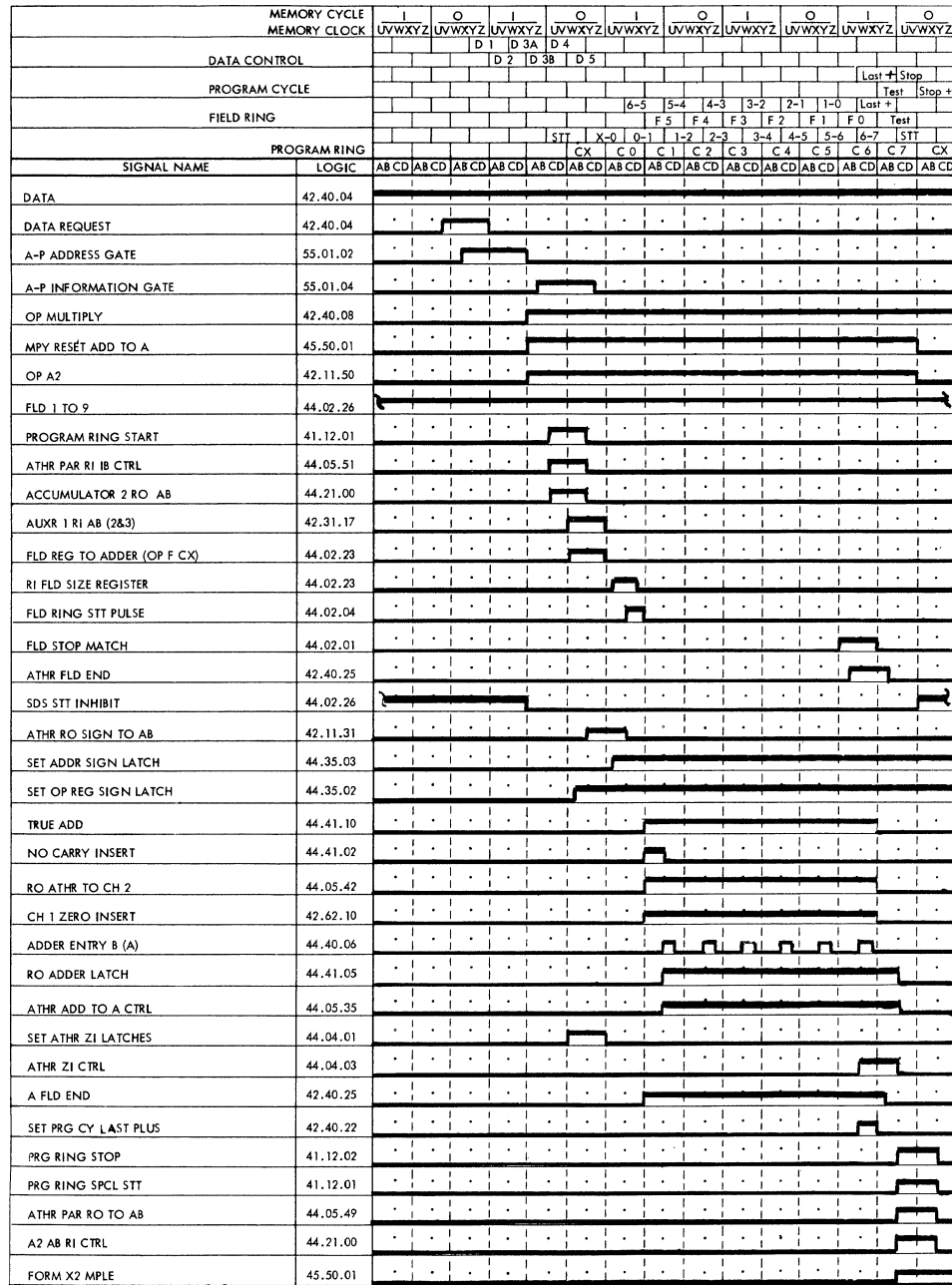


FIGURE 5.2-3. ZERO AND ADD TO A2--MULTIPLY OPERATION

B. Form Times Two Multiple (Figure 5.2-4)

The second step is started at the end of the first. The program ring is restarted and the second latch of the series (form X2 multiple) is set by the program cycle stop pulse. The first latch is reset with the start of the program ring.

At the start of the cycle, the multiplicand factor in accumulator 3 is parallel-transferred to the arithmetic register where it serves as the X1 multiple factor. During the transfer, the op'd A sign latches are reset and set to the sign of the factor. This sign value remains set until the end of the operation.

The adding operation is performed directly on the accumulator. The arithmetic and auxiliary registers and the field control feature are not used. Simultaneously, with its parallel read-out, accumulator 3 starts to serial shift to the right to read out of its output latches. Gates are brought up to read the data to both channel 1 and channel 2. The data is gated to the adder with true-add and a no-carry insert. The adder output is returned serially to the high-order position of accumulator 3. During the first serial transfer, the accumulator 3 high order reads in a zero from the adder output to serve as the eleventh digit to satisfy a possible carry condition. The eleven digits entering the accumulator are shifted across until the low order is in the output latch. Accumulator 3 now contains two times the value of the multiplicand.

During the adding cycle, accumulator 1 is serial shifted to the left with a zero insert at the low order. This clears the previous data from the register and provides a valid factor for the first multiply add cycle.

The program ring is allowed to run through C10. A recycle pulse causes the ring to restart at CX of the following digit time. The final digit from the adder output occurs during CX of the new cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 3 RO to Athr		
Op Sign Reset	CX	5H-44.35.10
A 3 AB RO Ctrl	CX-0	4G-44.31.00
Athr Par RI AB	CX-0	3F-5F-44.05.50
RI AB Op Sign	C0	3E-42.11.30
A 3 Serial Ctrl		
A 3 Sh Rt L Turn On	CX-0	4C-44.31.02
Blank Par Regen RI	C0	3D-44.31.06
A 3 RI From Adder	C0	5E-44.31.07
RO A 3 Ch 1	C0	4B-5B-44.31.09
RO A 3 Ch 2	C0	4F-5F-44.31.09
A 3 Sh Rt L Reset	CX-0 (cy 2)	4E-44.31.02

C. Form Times Four Multiple (Figure 5.2-5)

The third step of the operation is started at C10 by setting the form X4 multiple latch. The program ring is recycled and the form X2 multiple latch resets at CX.

The two times multiple in accumulator 3 must be added to itself to produce the four times multiple. Because the factor contains a possible eleven digits, it cannot be serially transferred with regeneration. The ten digits within the register are parallel-transferred to the auxiliary register to start the cycle. The adding operation is performed directly on the auxiliary register without the use of the arithmetic register and field control features. Simultaneously, with the parallel read-out, accumulator 3 output latches are read to both channel 1 and channel 2. The entire auxiliary register is serially shifted to the right to feed the auxiliary 3 output latches. Gates are brought up on the second digit time to feed the remaining ten digits to both channel 1 and channel 2. The data is gated to the adder with true-add and no-carry insert. The adder output is returned serially to the high-order position of auxiliary 1. The eleven digits entering the register are shifted across until the low order is in the output latch of auxiliary 3. The auxiliary register now contains four times the value of the multiplicand.

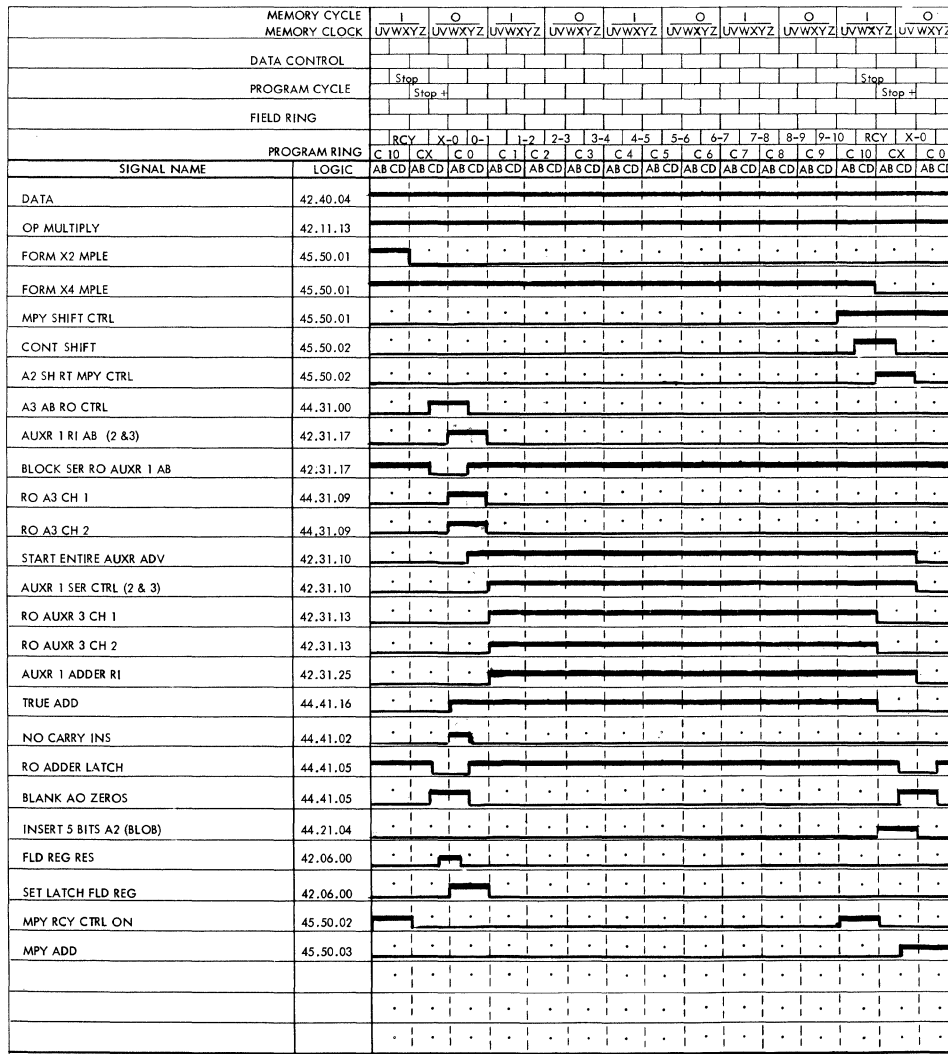


FIGURE 5.2-5. FORM X4 MULTIPLE

During the adding cycle, the field register is set to the full field value (09). The field ring is used to serially scan out the arithmetic register, when the one time multiple is required. The field register remains set for the rest of the operation. The program ring runs through C10. A recycle pulse causes the ring to restart at CX in the following digit time. The final digit from the adder output occurs at CX-time of the new cycle. At C10-time, the multiply shift control latch is set to start the actual multiply operation. The continue shift latch is in turn set at C10 CP, and the A 2 shift right multiply control at CX. These latches are used to control the shifting of accumulators 1 and 2, when a new multiplier factor is required. Normally, the low-order position of accumulator 1 (Pos 9) is shifted into accumulator 2. On the first shift, this entry is inhibited and the five-bit tag is inserted. The low order reads into the accumulator 2 output latches and the parallel output latches of the shift register. It is assumed at this point that the first digit is of significant value.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Par Transfer A 3 to Auxr		
A 3 AB RO Ctrl	CX-0	4H-44.31.00
Block Ser RO (Auxr)	CX-0	3D-42.31.17
Auxr 1 RI AB	C0	3D-5A-6A-42.31.17
Auxr 2 RI AB	C0	4D-5D-42.31.19
Auxr 3 RI AB	C0	4B-5B-42.31.21
RO Data to Adder		
RO A 3 Ch 1	C0	5D-44.31.09
RO A 3 Ch 2	C0	5E-44.31.09
Stt Entire Auxr Adv	C0-1	4C-42.31.10
Auxr 1 Ser Ctrl	C1	5C-5D-42.31.10
Adv Auxr 2	C1	4E-5D-42.31.11
Adv Auxr 3	C1	4B-5B-42.31.12
RO Auxr 3 Ch 1	C1	4C-5A-42.31.13
RO Auxr 3 Ch 2	C1	3E-4E-42.31.13
Auxr 1 Adder RI	C1	4F-5F-42.31.25
Stop Entire Auxr Adv	CX-0 AP (cy 2)	4H-4G-42.31.10
Adder Controls		
Turn on True-Add	C0	5C-44.41.16

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
No-Carry Insert	C0	44.41.02
RO Adder Latch	C0 CP	4A-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Set Field Register		
Fld Reg Reset	CX DP	2D-42.06.00
Set Latch Fld Reg	C0	4G-42.06.00
Fld Reg Set TP 1B	C0	3F-44.02.25
Fld Reg Set TP 2B	C0	4F-44.02.25
Fld Reg Set UP 3B	C0	5E-42.06.04
Fld Reg Set UP 6B	C0	5E-42.06.05
Mpy Rcy Ctrl On	C10	5H-45.50.02
Prg Cy Stop	C10	42.40.21
Insert 5-Bit Tag	CX (cy 2)	2G-44.21.04

D. Multiply Add - X4 Multiple (Figure 5.2-6)

Following the formation of the multiples, any one of the factors may be selected for the first multiply add cycle. The detection of a significant digit (non-zero) sets the multiply add latch. The actual factor is decoded to select the multiple. The digit 6 of the assumed multiplier (106) selects the X4 multiple first.

The X4 multiple is in the auxiliary register and is assumed to contain a possible eleven digits. Because the factor cannot be serially read out and regenerated with a digit in the output latches, the register portion is parallel-transferred to the word buffer register. The low-order digit is read from the auxiliary 3 output latches. This is followed by a serial read-out of the remaining ten digits in the word buffer register. Simultaneously, accumulator 1 is serial shifted to the right to read-out through its output latches. A zero is read into the high-order position on the first shift to serve as the eleventh digit. The data at the output latches is gated to channel 2. Both channels are read to their respective adder entries with true-add and no-carry insert. The adder output serially enters the high-order position of accumulator 1. The eleven-digit total is shifted across the register until the low order enters the output latch, provided the multiplier digit has not been reduced to zero.

During the multiply add cycle, the multiplier digit is read into the shift register with a shift right control. The digit value is reduced by the value of the multiple by

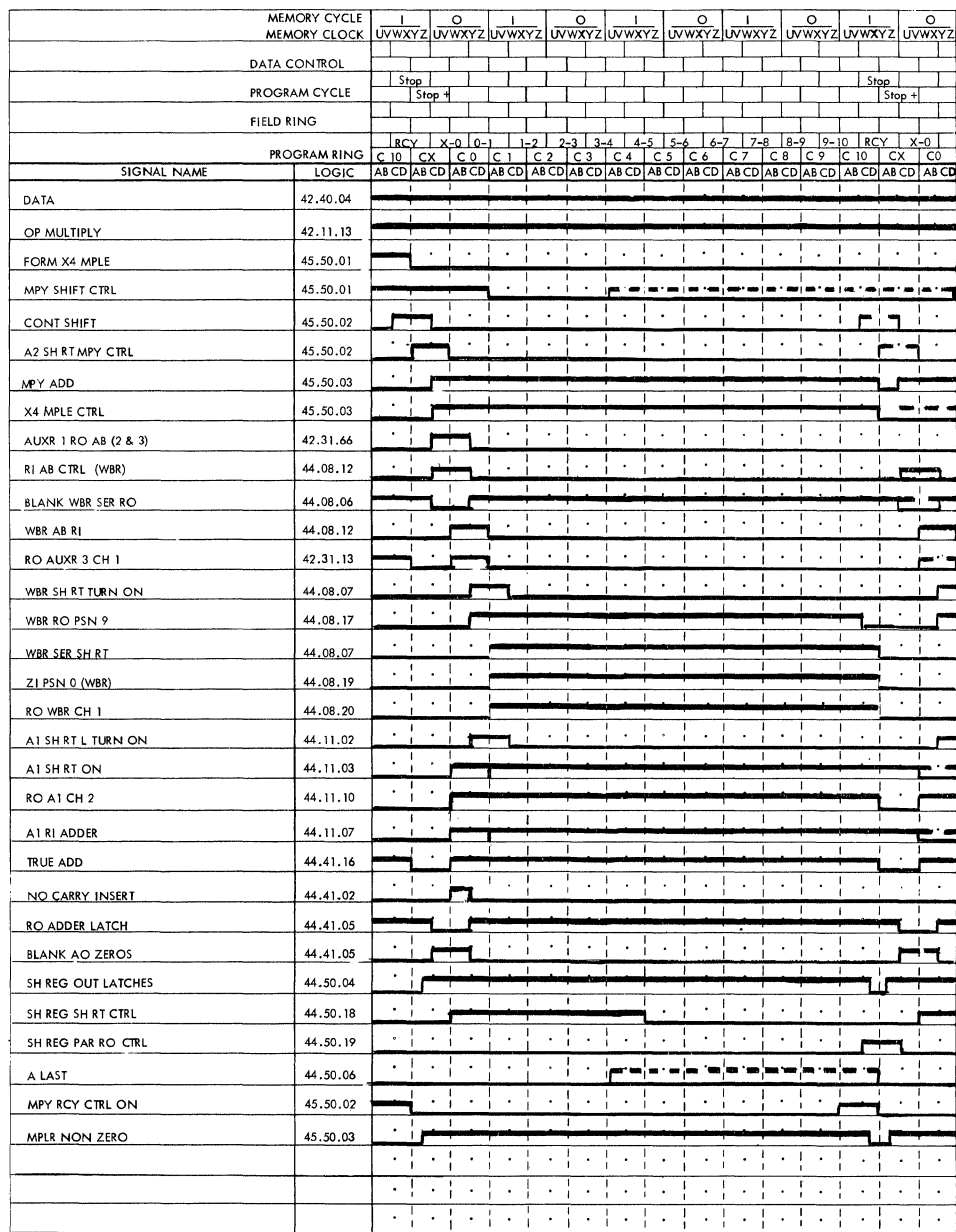


FIGURE 5.2-6. X4 MULTIPLE CONTROL

further right shifting. At the end of the adding cycle, the shift register is read out to its parallel output latches for the next selection. The reduction of the digit value to zero is discussed in the following section.

The program ring runs through C10 with a restart at CX of the following digit time. The final adder output digit is again read in during CX of the following cycle.

Major Signals

Key Timing

Logic and Location

Multiply Control

Mpy Add

CX CP

4C-45.50.03

X4 Mple Ctrl (6B)

CX CP

6B-45.50.03

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Transfer Auxr to WBR		
Auxr 1 RO AB	CX-0	4G-42.31.66
Auxr 2 RO AB	CX-0	3G-42.31.19
Auxr 3 RO AB	CX-0	5A-42.31.21
RI AB Ctrl (WBR)	CX-0	44.08.12
Blank WBR Ser RO	CX-0	2J-44.08.06
WBR AB RI	C0	4B-5B-44.08.12
RO Mple to Adder		
RO Auxr 3 to Ch 1	C0	5F-42.31.13
WBR Sh Rt Turn On	C0-1	5A-44.08.07
WBR RO Psn 9	C0-1	4B-44.08.17
WBR Ser Sh Rt	C1	5D-5B-44.08.07
ZI Psn 0 (WBR)	C1	44.08.19
RO WBR Ch 1	C1	5J-44.08.20
Reset WBR Sh Rt	CX (cy2)	3E-4C-5B-44.08.07
RO A 1 to Adder		
A 1 Sh Rt L Turn On	C1	2A-44.11.02
RO A 1 Ch 2	C0	4F-5F-44.11.10
A 1 RI Adder	C0 (C1)	3A-44.11.07
A 1 Sh Rt L Reset	C0 (cy2)	3D-44.11.02
Adder Controls		
Turn On True-Add	C0	5B-44.41.16
No-Carry Ins	C0	44.41.02
RO Add Latch	C0 CP	4A-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Shift Register Controls		
Block Ser RO Ctrl (RI)	CX-0	4J-44.50.19
Ser RO Ctrl	C0 CP	4F-44.50.19
Sh Rt Turn On	C0	5A-44.50.18
Sh Rt Reset	C5	3C-44.50.18
Sh Reg Par RO Ctrl	PC Stop	5D-44.50.19
Mpy Rcy Ctrl On	C10	5H-45.50.02
Prg Cy Stop	C10	42.40.21

E. Multiply Add - X2 Multiple (Figure 5.2-7)

Selection of the second or third multiple for a multiplicand digit entails no special circuits. The remaining part of the digit is read from the shift register into its parallel output latches. The presence of a significant digit sets the multiply add latch again. The bit combination determines the multiple to be used. In the assumed problem, the original 6 has been reduced by 4, leaving a 2. This causes the X2 multiple cycle to be selected.

The X2 multiple is in accumulator 3 and is assumed to contain a possible eleven positions. Like the X4 multiple, the factor cannot be serially transferred directly from the register. The ten digits within the register are parallel-transferred to the word buffer register. Simultaneously, the low-order digit is read from the accumulator 3 output latches to channel 1. The word buffer register then starts a serial shift to the right to read out the remaining ten digits to channel 1. The eleven digits of accumulator 1 and its output latches are serially read out to channel 2 in time with the digits on channel 1. The data is gated from the channel to the adder entries with true-add and no-carry insert.

The adder output is serially entered into the high-order position of accumulator 1. With the assumed factor, the digit has been reduced to zero and the multiply shift latch is set. As the eleven digits shift across accumulator 1, the gating is changed to read the low order into the high-order position of accumulator 2. Even though the digit still reads into accumulator 1 output latches, it serves no purpose. The digits in accumulator 2 shift to the right to read out a new multiplier digit.

The operation of the shift register differs on the last selected multiple of a digit. As the remainder is shifted to count out the value of the multiple, the last shift moves the bit to position 9 (zero value) and reads out to set the A last latch. The signal from the A last latch, in turn, sets the multiply shift latch. At the end of the cycle, the continue shift and A2 shift right Mpy Ctrl latches are set. The former resets the output latches of both accumulators 1 and 2, while the latter controls the shifting for a new multiplier digit. The read-out of a zero multiplier digit causes the shifting to continue until a significant digit is sensed. The zero sensing and the resulting multiple idle cycle are discussed in a later section.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A3 AB RO Ctrl	CX-0	4J-44.31.00
RI AB Ctrl (WBR)	CX-0	44.08.12
Blank WBR Ser RO	CX-0	2J-44.08.06
WBR AB RI	C0	4B-5B-44.08.12
RO Mple to Adder		
RO A3 Ch 1	C0	5A-44.31.09
WBR Sh Rt Turn On	C0-1	4A-44.08.07
WBR RO Psn 9	C0-1	4B-44.08.17
WBR Ser Sh Rt	C1	5D-5B-44.08.07
ZI Psn 0 WBR	C1	44.08.19
RO WBR Ch 1	C1	5J-44.08.20
Reset WBR Sh Rt	CX (cy2)	3E-4C-5B-44.08.07
RO A1 to Adder		
A 1 Sh Rt L Turn On	C1	2A-44.11.02
RO A1 Ch 2	C0	4F-5F-44.11.10
A1 RI Adder	C1	3A-44.11.07
A1 Sh Rt L Reset	C0(cy2)	3D-44.11.02
Adder Controls		
Turn on True-Add	C0	5B-44.41.16
No-Carry Ins	C0	44.41.02
RO Adder Latch	C0 CP	4A-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Shift Register Controls		
Block Ser RO Ctrl (RI)	CX-0	4J-44.50.19
Ser RO Ctrl	C0 CP	4F-44.50.19

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh Rt Turn On	C0	5A-44.50.18
Sh Rt Reset	C3	3B-44.50.18
A Last	C4 AP	4B-5B-44.50.06
A Last +	C4 CP	4G-5G-44.50.06
Multiply Shift Control		
Mpy Shift Ctrl	C4 CP	3J-4G-45.50.01
Cont Shift	C10 CP	3A-4A-45.50.02
A2 Sh Rt Mpy Ctrl	CX (cy2)	3D-4D-45.50.02
Mpy Rcy Ctrl On	C10	5H-45.50.02
Prg Cy Stop	C10	42.40.21

F. Multiply Idle - Zero Factor Digits (Figures 5.2-8 and 5.2-9)

The assumed multiplier factor (106) places a zero in the output latches on the next shift. A zero is first sensed at the accumulator 2 output latches, where it is used to set the continue shift latch for another digit. In turn, the A2 shift right Mpy Ctrl latch is set to read out the next digit by shifting both accumulators 1 and 2. If several zeros are read out in succession, the process is repeated until a significant digit is reached. The multiply shift latch is reset at C1 regardless of the digit read out. Each multiplier digit read out is read into the parallel output latches of the shift register to be analyzed for a non-zero entry. A significant digit reading out at the normal CX-time sets the multiply add latch for the cycle. When a zero is present, the time for the additional shift makes it impossible to start a multiply add cycle until the program ring is recycled. When a significant digit is detected after a zero, the non-zero condition sets the multiply idle latch to recycle the program ring. At CX-time of the new cycle, the multiply add latch can be set in the normal manner and the operation continued. The program ring runs a minimum of three digits for an idle cycle with an additional digit time for each zero.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Cont Shift	CX CP	3B-4A-45.50.02
A2 Sh Rt Mpy Ctrl	C0 AP	3D-4D-45.50.02
Mplr Non-Zero	C0 AP	2X-45.50.03
Mpy Idle	C0 CP	2F-3F-45.50.02
Mpy Rcy Ctrl On	C1 AP	5G-45.50.02
Prg Cy Stop	C1	42.40.21

		MEMORY CYCLE MEMORY CLOCK		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ		0 UVWXYZ		1 UVWXYZ			
DATA CONTROL																															
PROGRAM CYCLE		Stop Stop + Stop Stop + Last + Stop																													
FIELD RING		I 9 I 9-8 8-7 7-6 6-5 5-4 4-3 3-2 2-1 1-0 F 9 F 8 F 7 F 6 F 5 F 4 F 3 F 2 F 1 F 0 Test																													
PROGRAM RING		RCY X-0 0-1 RCY X-0 0-1 1-2 2-3 3-4 4-5 5-6 6-7 7-8 8-9 9-10 RCY X-0 0-1 C 10 CX C 0 C 1 C X C 0 C 1 C 2 C 3 C 4 C 5 C 6 C 7 C 8 C 9 C 10 CX C 0 C 1 AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD																													
SIGNAL NAME		LOGIC																													
DATA		42.40.04																													
OP MULTIPLY		42.11.13																													
MPY SHIFT CTRL		45.50.01																													
CONT SHIFT		45.50.02																													
A2 SH RT MPY CTRL		45.50.02																													
MPY IDLE		45.50.03																													
MPY ADD		45.50.03																													
X2 MPLE CTRL		45.50.03																													
X1 MPLE CTRL		45.50.03																													
FLD RING STT		44.02.04																													
FLD RING RUN		44.02.04																													
FLD STOP MATCH		44.02.01																													
FLD RING LAST		44.02.02																													
FLD CHECK LATCH		44.02.24																													
ATHR SCAN MATRIX		44.01.11																													
RO ATHR CH 1		44.05.41																													
CH 1 INS 0		42.62.10																													
A1 SH RT L TURN ON		44.11.02																													
A1 SH RT ON		44.11.03																													
RO A1 CH 2		44.11.10																													
A1 RI ADDER		44.11.07																													
TRUE ADD		44.41.16																													
NO CARRY INSERT		44.41.02																													
RO ADDER LATCH		44.41.05																													
BLANK AO ZEROS		44.41.05																													
SH REG OUT LATCHES		44.50.04																													
SH REG SH RT CTRL		44.50.18																													
SH REG PAR RO CTRL		44.50.19																													
A LAST		44.50.06																													
MPY RCY CTRL ON		45.50.02																													
MPLR NON ZERO		45.50.03																													

FIGURE 5.2-8. MULTIPLY IDLE--X1 MULTIPLE CONTROL

G. Multiply Add - X1 Multiple (Figure 5.2-8)

In the assumed problem, a one has been read into the output latches of the shift register. The multiply add latch is set and the X1 multiple control is brought up by the 0-1 bit combination. The basic operation is the same with or without a zero ahead or when the selection is part of a larger factor. The difference is in the control of accumulator 1 to position the factor for the eleven-digit add cycle. If the X1 selection is part of a larger digit, the previous cycle leaves eleven digits in the register and its latches. When the X1 or any multiple is the first selection of a digit, the accumulator 1 factor must be shifted to the right to fill the output latches. With no zero ahead, the multiply shift signal gates the shift at CX. A zero delays the operation until after the multiply shift latch has been reset. The shift is then gated by the multiply idle signal at C0. In either case, a zero is inserted at the high order of accumulator 1 from the adder output channel.

The X1 multiple is in the arithmetic register. It contains a maximum of ten digits, and thus, must have a zero inserted in the high-order position during the transfer. The field register was set to full field (0-9) during the third step. The field ring is set from the field register in the normal manner. The field length check is not used because of the lack of time in the cycle. The check latch is forced.

The X1 multiple reads out serially from the arithmetic register as the field ring advances and is fed to channel 1. The channel 1 zero insert is used to supply the eleventh digit. Simultaneously, the eleven digits of accumulator 1 and its output latches are read out to channel 2. The channels are gated to the adder entries with true-add and no-carry insert. The adder output is serially read into the high order of A1. The data is shifted until the low order reaches the output latch. The X1 multiple cycle is always a last selection, and thus, the low order is also shifted into accumulator 2 and a new digit is shifted out.

The program ring again runs through C10 and is recycled. The final adder output and the shift of accumulator 2 occur at CX of the new cycle. The multiply shift was again set as the result of the shift register going to zero.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Multiply Control		
Mpy Add	CX CP	4C-45.50.03
1X Mple Ctrl (1B-0B)	CX CP	6G-45.50.03
Field Ring Control		
Fld Ring Stt Pulse	CX CP	3D-5A-44.02.04
Set Fld Ring Stt Pos (F1-9)	CX CP	3A-5A-44.03.01
Fld Ring Run	C0	5D-44.02.04
Fld Stop Match (P0)	F0	4J-44.02.01

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Fld Ring Last	BP	3A-4A-44.02.02
Fld Ring Chk	F Last	5F-44.02.22
RO Mple to Adder		
Athr Scan Matrix	F 1/2 Time	XX-44.01.11
Athr Output Latches	AP	5E-44.01.16
RO Athr Ch 1	C0	3A-2B-44.05.41
Reset Athr Ch 1 L	C10	2C-3C-44.05.41
Ch 1 Ins 0	C10	5F-42.62.10
RO A1 to Adder		
A1 Sh Rt L Turn On	C1	2A-44.11.02
RO A1 Ch 2	C0	4F-5F-44.11.10
A1 RI Adder	C1	3A-44.11.07
A1 Sh Rt L Reset	C0 (cy2)	3D-44.11.02
Adder Controls		
Turn On True-Add	C0	5C-44.41.16
No-Carry Ins	C0	44.41.02
RO Adder Latch	C0 CP	4A-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Shift Register Controls		
Block Ser RO Ctrl (RI)	CX-0	4J-44.50.19
Ser RO Ctrl	C0 CP	4F-44.50.19
Sh Rt Turn On	C0	5A-44.50.18
Sh Rt Reset	C2	3A-44.50.18
A Last	C3 AP	4B-5B-44.50.06
A Last +	C3 CP	4G-5G-44.50.06

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Multiply Shift Control		
Mpy Shift Ctrl	C3 CP	3J-4G-45.50.01
Cont Shift	C10 CP	3A-4A-45.50.02
A2 Sh Rt Mpy Ctrl	CX (cy 2)	3D-4D-45.50.02
Mpy Rcy Ctrl On	C10	5H-45.50.02
Prg Cy Stop	C10	42.40.21

H. End Multiply - Sign Adjust (Figure 5.2-9)

The multiply add and multiply shift cycles are applied as the multiplier factor indicates. As accumulator 2 shifts, the five-bit tag moves across the register until it reads out to the output latches. This indicates that the actual multiplication is completed. In the assumed problem, the remainder of the multiplier digits are zeros. This causes the continue shift and the A2 shift right mpy ctrl latches to remain on. Both accumulator 1 and accumulator 2 right shift with zeros being inserted into accumulator 1 from the adder output channel. When the tag reads into accumulator 2 and the shift register output latches, it appears to be a significant digit or non-zero. The multiplier non-zero signal sets the multiply idle latch to cause a program ring re-cycle. The multiply idle latch reset is forced by the set of the multiply end latch to prevent a shift of accumulator 1 at C0-time.

The presence of the five-bit tag in the accumulator 2 output latches sets the multiply end latch. This, in turn, prevents the setting of the multiply add latch and the selection of a multiple for a new cycle.

The original multiplicand factor is in the arithmetic register serving as the X1 multiple. It is parallel-transferred to accumulator 3 for future use. The multiplier factor has been destroyed, but still exists in core storage at the location originally specified by program register D. Accumulators 1 and 2 now contain up to a twenty-digit product for future use. Before the end of the operation, each of the accumulators is parallel-transferred to the arithmetic register for sign adjustment. The signs of the two factors were stored during the first two steps of the operation. The algebraic result is inserted while the accumulator data is in the arithmetic register. Accumulator 1 is changed first. If one of the factor signs is alpha, the product is set alpha.

The end data operation latch is set during C5 of the multiply end cycle. In turn, the data latch is reset and the program cycle ring is started at the Test position to stop the program ring after C6. The multiply operation signal goes down after the instruction control starts, which, in turn, resets the remaining multiply controls.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Mpy	PC Stop	2F-3G-4F-45.50.03
Set Signs Plus (Minus)	End Mpy	3C-44.35.67

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Return Multiplicand		
Athr Par RO AB	CX-0	4D-44.05.49
A3 AB RI Ctrl	CX-0	4B-44.31.00
Adjust A1 Sign		
A1 RO AB	C0-1	4E-44.11.01
Athr Par RI AB Ctrl	C0-1	3E-44.05.50
Sign Chg Ctrl	C1-2	5H-44.35.65
Athr Par RO AB	C2-3	4E-44.05.49
A1 AB RI Ctrl	C2-3	2G-44.11.00
Adjust A 2 Sign		
A 2 AB RO Ctrl	C3-4	4B-44.21.00
Athr Par RI Ctrl	C3-4	3D-44.05.50
Sign Chg Ctrl	C4-5	5J-44.35.65
Athr Par RO AB	C5-6	4F-44.05.49
A 2 AB RI Ctrl	C5-6	3F-44.21.00
End Operation		
End Data Op	C5-6	3J-42.40.30
Set Prg Cy Test	C6	3A-5A-42.40.21
Prg Ring Stop	PC Test CP	3F-4G-5G-41.12.02

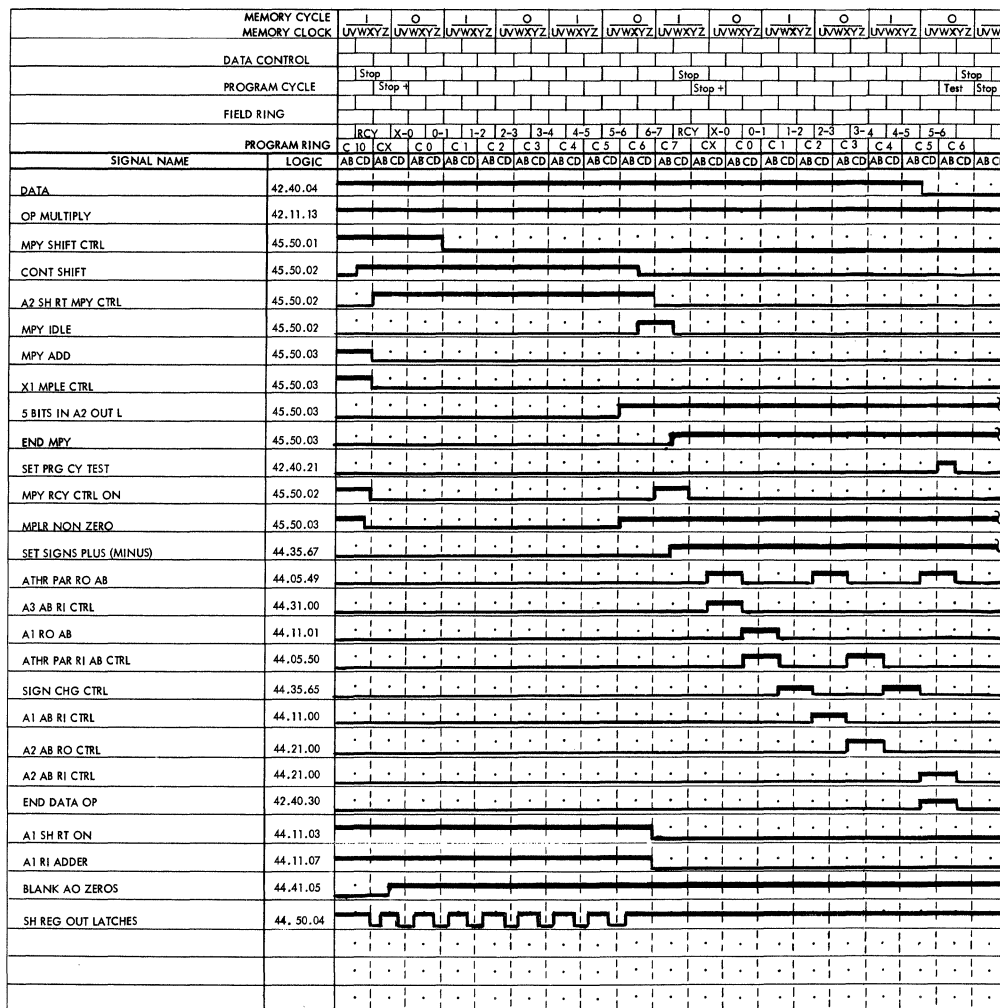


FIGURE 5.2-9. MULTIPLY IDLE--MULTIPLY END

FIGURE 5.2-10. PRODUCT DEVELOPMENT DURING MULTIPLICATION

CYCLE	ACCUMULATOR 1	OL	ACCUMULATOR 2	FACTORS	REMARKS
START	XXXXXXXXXX		XXXXXXXXXX	5986237401	IN A3
RESET ADD	XXXXXXXXXX		0000725108	5986237401	X 1 A THR
FORM X2	0000000000		0000725108	11972474802	X 2 A 3
FORM X4	0000000000		0000725108	23944949604	X 4 AUXR
8 MPY SHIFT	0 0000000000	0	* 000072510 8 8		A 1 SHIFT
X 4	2394494960	4	* 000072510 8 4	23944949604	ADD X 4
X 4	4788989920	8	8*00007251 0 0	23944949604	ADD X 4
0 CONT SH	0 478898992	0	08*0000725 1 1		SKIP ZERO
1 MPY IDLE	0 047889899	2	08*0000725 1 1		A 1 SHIFT
X 1	0 646513639	3*	308*000072 5 5	5986237401	ADD X 1
5 MPY SHIFT	0 064651363	9	308*000072 5 5		A 1 SHIFT
X 4	2 459146324	3	308*000072 5 1	23944949604	ADD X 4
X 1	3 057770064	4	4308*00007 2 2	5986237401	ADD X 1
2 MPY SHIFT	0 305777006	4	4308*00007 2 2		A 1 SHIFT
X 2	1 503024486	6	64308*0000 7 7	11972474802	ADD X 2
7 MPY SHIFT	0 150302448	6	64308*0000 7 7		A 1 SHIFT
X 4	2 544797409	0	64308*0000 7 3	23944949604	ADD X 4
X 2	3 742044889	2	64308*0000 7 1	11972474802	ADD X 2
X 1	4 340668629	9	364308*000 0 0	5986237401	ADD X 1
0 CONT SH	0 434066862	9	9364308*00 0 0		SKIP ZERO
0 CONT SH	0 043406686	2	29364308*0 0 0		SKIP ZERO
0 CONT SH	0 004340668	6	629364308* 0 0		SKIP ZERO
0 CONT SH	0 000434066	8	8629364308 * *		SKIP ZERO
END MULTIPLY	0 000434066		8629364308	5986237401	TO A 3 (X 1)

X = INSERT

X = UNUSED DIGIT

MULTIPLIER 0000725108
MULTIPLICAND 5986237401
PRODUCT 00004340668629364308

FIGURE 5.2-11A. MAJOR TIMING FOR PRODUCT DEVELOPMENT

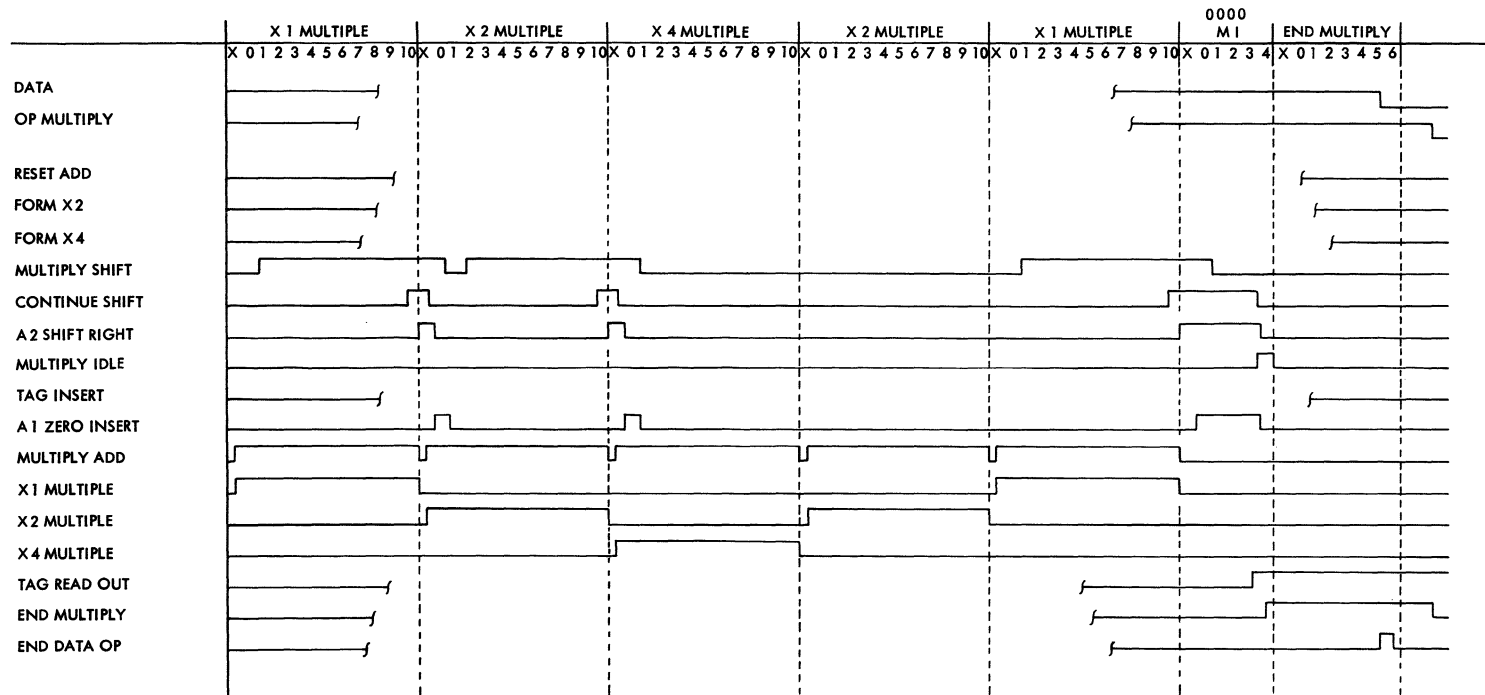
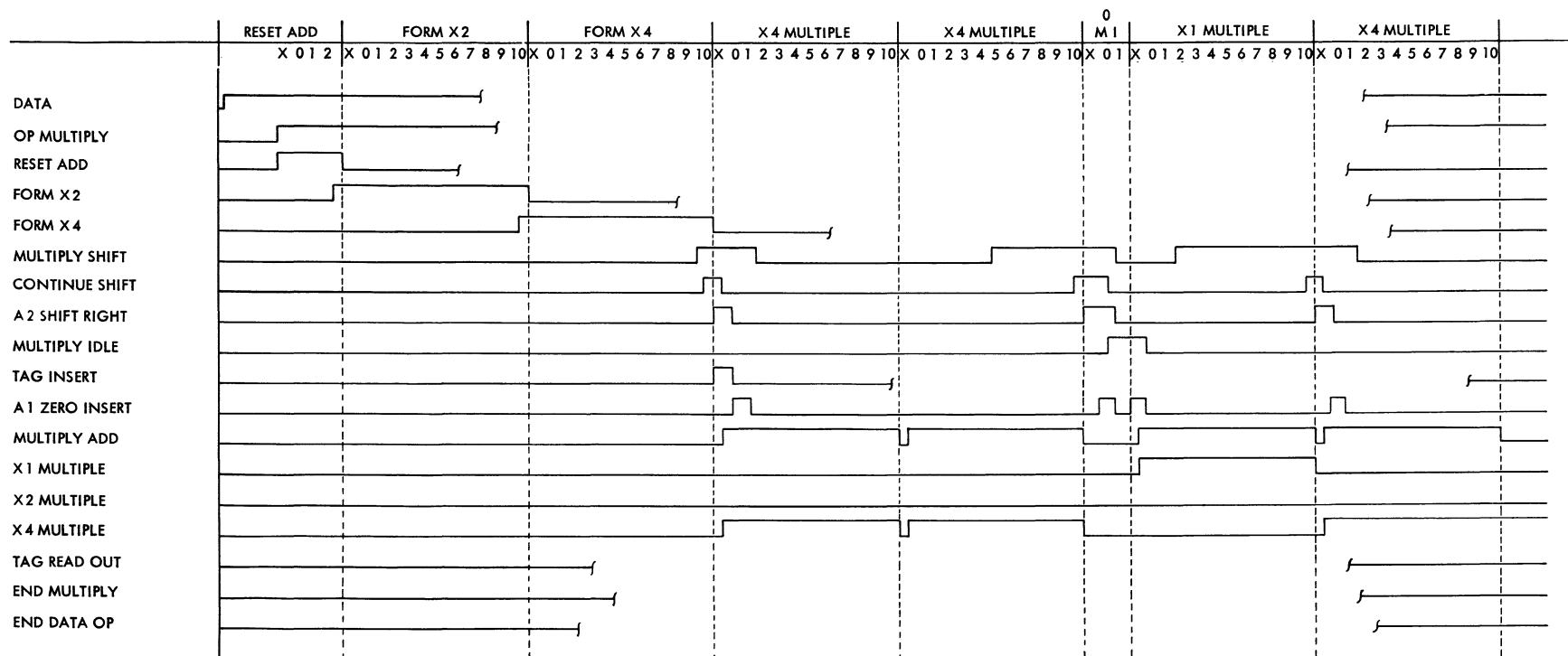


FIGURE 5.2-11B. MAJOR TIMING FOR PRODUCT DEVELOPMENT



5.3.00 DIVIDE (-53) - INTRODUCTION

Division is accomplished by an over and over reduction or subtraction of the divisor from the dividend. In practice, the process is effectively reversed. The dividend is complemented to make it a negative value and the divisor is added. The reversal of the process allows a zero value to be sensed as positive. The divisor is repeatedly added to the high-order position until an overdraw is sensed and then moved to the right to repeat the process. Each divide-add cycle, which does not produce an overdraw, adds one to the appropriate position of the quotient. Overdraw correction is accomplished by retaining the working portion of the dividend in a separate register to be re-entered in case of an overdraw.

Rule

A dividend of up to twenty digits is placed in accumulators 1 and 2 prior to the divide operation. Accumulator 1 is the high order. A divisor of up to ten digits is entered into accumulator 3 from the data address of the instruction. The factor enters similar to a zero and add operation. If the field register reads "09", the data word is parallel-transferred through the arithmetic register without processing. With a shorter field, the factor is selected and located in the arithmetic register by an adder cycle before transfer to accumulator 3 on the arithmetic bus. A ten-digit quotient is developed in accumulator 2 with a remainder of up to ten digits left in accumulator 1. The absolute value of the divisor must be greater than the absolute value of the portion of the dividend in accumulator 1. This is required to prevent attempts to develop a quotient digit greater than nine. A divisor that is too small or zero will cause an error as the first quotient digit passes nine.

Divide Setup

The operation starts by complementing first accumulator 2, then accumulator 1, by a right shift serial transfer as one operation. They are read onto channel 2 to adder entry B with complement-add control. The normal carry insert is blanked on the first digit to form a nine's complement. Zeros are entered on channel 1 to adder entry A from the channel digit insert. The adder output is fed serially directly to the high order of the accumulators. The first nine digits are read into accumulator 2 and a five-bit tag is inserted as the tenth digit in the high order. The tag will be used to signal the end of operation. The remaining eleven digits from the adder output are entered into accumulator 1 with the low order digit shifted into the output latches. Figure 5.3-1 shows the data path for the complementing operation.

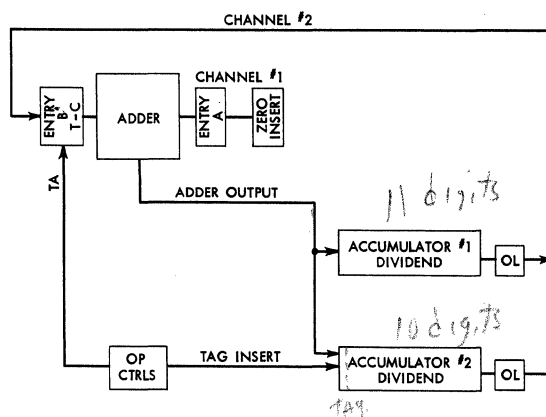


FIGURE 5.3-1. DIVIDEND COMPLEMENT DATA FLOW

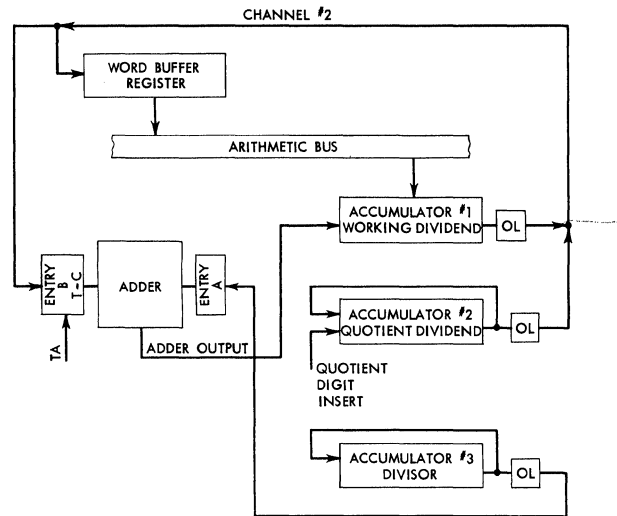


FIGURE 5.3-2. DIVIDEND REDUCTION DATA FLOW

Reduction Cycle

Accumulator 1 is read out serially to channel 2 starting with the output latches. This data is fed both to adder entry B with true-add control and to the high order of the word buffer register. Only the ten low-order digits are read into the word buffer register. Accumulator 3 is serially read out to channel 1 and adder entry A, while the register is regenerated. A zero is inserted for the eleventh or high order position by the channel 1 digit insert. The eleven-digit adder output serially returns to accumulator 1 reduced by the value of the divisor.

As the last digit leaves the adder on each reduction cycle, the carry outputs are tested. If no carry occurs, it indicates the working dividend was larger than the divisor and the operation should continue with another reduction. Figure 5.3-2 shows the data flow paths for both the reduction and overdraw correction.

Overdraw Correction

When the last adder output indicates a carry, the divisor was greater than the working dividend and has reversed its sign or overdrawn. The word buffer register contains the working dividend prior to the last operation except for the high-order position. Because the divisor could not be added again, the high-order digit must have been a nine or complement zero and, thus, can be ignored.

The word buffer register is parallel-transferred to accumulator 1 by the arithmetic bus. These ten digits now rest in the high-order ten positions to effect a one position left shift for the next operation. The digit value in the output latches of accumulator 1 is now in error and is not used.

Accumulator 2 Shift

During the first reduction cycle of each quotient digit, accumulator 2 is right shifted nine positions and regenerated to effect a one position left shift. The use of right shift allows leaving the next high order digit in the output latches at the end of each rotation. This last digit to the output latches is not regenerated in the high order position, leaving it blank for subsequent entry of the quotient digit.

On the first reduction cycle, after each overdraw, accumulator 2 output latches are read out to channel 2 first and then are followed by the ten digits within the accumulator 1 register for the eleven-digit working dividend.

Quotient Development

The shift register is used to keep count of the reduction cycles to determine the value of the quotient digit to be entered. The shift register is set to zero on the first reduction cycle of each group. For each adder no-carry cycle, the shift register is advanced or shifted by one. In the case of an overdraw, the cycle is not counted. Immediately after an overdraw is sensed and before accumulator 2 starts its shift cycle, the shift register reads out through its parallel output latches. This digit is inserted into the position left blank on the previous accumulator 2 shift cycle.

End Divide

When the five bit tag has moved across accumulator 2 and is read out to the output latches, the end divide circuits are conditioned. Reduction cycles continue to develop the tenth quotient digit. The tenth, or last quotient digit, is entered into the high-order position of accumulator 2 on the subsequent overdraw correction. Accumulator 2 takes one more shift cycle to bring the last entered quotient digit into the low order position (pos. 9).

The correct remainder, after the last overdraw, is parallel-transferred from the word buffer register to accumulator 1. The remainder is still in complement form and must be serially read out to channel 2 and adder entry B with complement-add control. The normal carry is again blocked to produce a 9's complement. Zeros are fed to adder entry A from the channel 1 digit insert. The adder output is returned to accumulator 1 high order to form a true remainder. Use of the quotient in accumulator 2 and the remainder in accumulator 1 require additional instructions on subsequent program steps.

Sign

The sign of the quotient in a divide operation is normally determined algebraically. If the signs of the divisor and dividend are alike, the quotient is plus. If the factor signs are different, the quotient is minus. An exception occurs if either factor sign is alpha, then the quotient is set alpha.

The sign is determined by the setting of the sign latches on the arithmetic bus. The dividend sign was set by parallel-transferring accumulator 1 to the word buffer register during the accumulator 2 complement time. The sign of the divisor was set during the zero and add to accumulator 3. During the complementing of accumulator 1, accumulator 2 was parallel-transferred to the arithmetic register and returned to set the quotient sign.

The remainder sign is that of the dividend in accumulator 1.

5.3.01 Divide

The divide operation is performed by simple over and over reduction of the dividend by the divisor. With the exception of the controls, all of the steps to perform

the operation are simplified add cycles. The cycles include reading in the divisor factor, complementing the dividend, adding the divisor, and recomplementing the remainder. The dividend is placed in accumulators 1 and 2 prior to the start of the operation. The steps are selected and sequenced by a group of divide operation latches, which provide the necessary operation gates.

Two basic cycles are used to develop the quotient. Both are divide-add, or reduction, cycles. The divide left shift add is used on the first cycle of a quotient digit. It controls the necessary gates to insert the previous quotient digit and to shift the factors to start the development of the next digit. The continue divide add cycle is used for all subsequent cycles in which it is only necessary to count the number of cycles as the reduction is made. The selection of the cycle to be used is dependent on the adder output carry condition indicating an overdraw. A third divide add control, known as divide start, is similar to a divide left shift add and is used to start the operation.

The shift register is used to count the divide-add cycles. During the first reduction cycle of any quotient digit, the parallel output latches of the shift register are set to zero (2 & 1 bit latch on). The output latches are read into the shift register cores with a left shift which sets position 8. The output latches retain the zero setting.

During the second reduction cycle, position 8 is read out to the output latches causing the one and zero bit latches to be turned on. The output latches are again read into the shift register cores with a left shift which sets position 7. The output latches now contain a one, and the cores have been set to produce a two on the next reduction cycle. When an overdraw occurs, the value in the output latches is inserted in accumulator 2 as the quotient digit. This value does not include the count of the reduction cycle which produced the overdraw.

The operation is detailed by steps or cycles. No specific problem is assumed in the description of the operation. The timing charts (Figures 5.3-3 to 8) show the time relations of each of the cycles. The alternate conditions for the next cycle are shown dotted to allow signal tracing for any sequence. The reduction of a simple dividend factor is shown in Figure 5.3-9, and a simplified timing chart for the same problem is shown in Figure 5.3-10. By reference to the detailed charts above for each cycle, the entire operation can be visualized.

The following major cycles are used to perform a divide operation (Each is detailed below):

- A. Zero and Add to Accumulator 3
- B. Complement Accumulators 2 and 1 - Sign Adjust
- C. Divide Start Cycle
- D. Continue Divide Add Cycle
- E. Divide Left Shift Add Cycle
- F. End Divide - Complement Remainder

- A. Zero and Add to Accumulator 3 (Figure 5.3-3)

The divisor factor comes from the location specified by positions 6-9 of the instruction. It enters the accumulator in exactly the same manner as in a zero and add operation. Field control is used and either the full word or any portion of the word may be specified. When less than the full word is indicated, the factor is moved

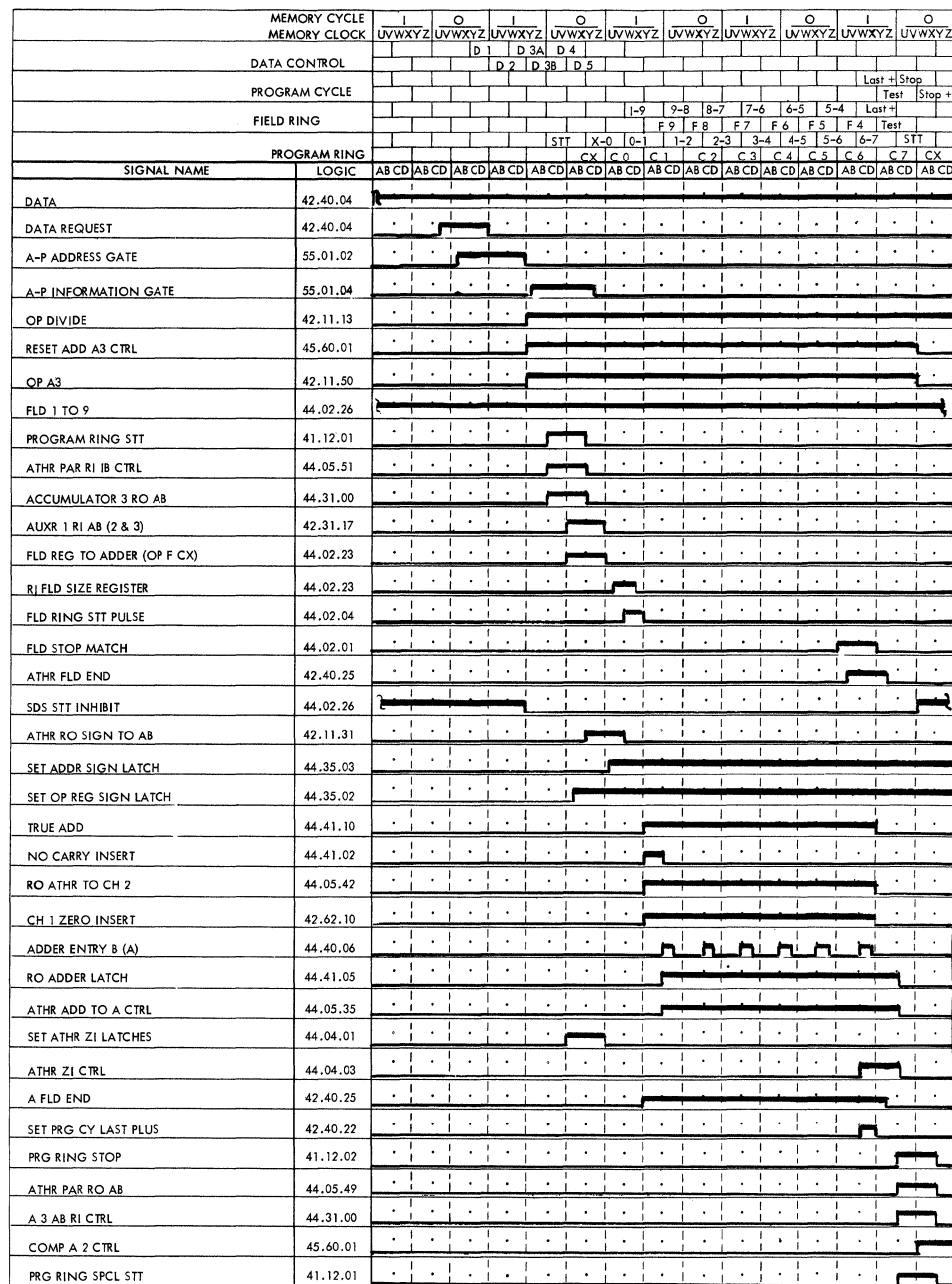


FIGURE 5.3-3. ZERO AND ADD TO A3--DIVIDE OPERATION

to the low-order positions. Care must be taken to insure that the divisor is larger than the working dividend (A1 plus high order of A2). Data previously in accumulator 3 is lost.

The first cycle of the operation is setup as soon as the divide op code is developed. The setting of the reset add A3 latch provides the signal to form op A3, add to A, reset add to A, and op matrix true-add. These are the normal requirements for a zero and add. The remaining signals developed from these controls are the same as for the basic operation described in Section 5.1.02.

The first step in sign control is performed as a part of the basic zero and add. The sign of the divisor factor enters the address sign latches while it is in the arithmetic register. These latches remain set throughout the operation. As in zero and add, the operated accumulator sign is set in the Op'd A sign latches. This sign is not used and is later reset. The sign entering accumulator 3 is that of the data address word.

The program ring is allowed to run normal in this cycle with a program cycle turn-off. The program cycle stop pulse is used to force a special start for the next step. The effect is the same as a recycle control with CX immediately following the last pulse of the first cycle.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Op Divide	DC3B	4D-42.11.13
Reset Add A3 Ctrl	DC3B	3C-4C-45.60.01
Op A3	DC3B	42.11.50
Op Reset Add to A	DC3B	42.11.05
Op Add to A	DC3B	42.11.05
Op Matrix True-Add	DC3B	42.11.09
Basic Reset Add Signals	See Section 5.1.02	

B. Complement Accumulators 1 and 2 - Sign Adjust (Figure 5.3-4)

The complementing of the dividend factor at the start operation is actually done in two steps. While separate control gates are used for each of the accumulators, the process is a continuous one. For the purposes of this discussion, the two steps are considered as one.

At program cycle stop of the zero and add cycle, the complement accumulator 2 latch is set to start the complementing. The reset add to A3 latch is reset at CX of the new cycle. The first cycle continues to C10, when the program ring is recycled. On the second cycle, the ring is recycled at C7. The complement A2 latch is reset at CX of the second cycle, while the complement A1 latch is set during C10 of the first cycle. The break in the transfer from one register to the other occurs at C9. The registers are controlled by the gates which are available at the time.

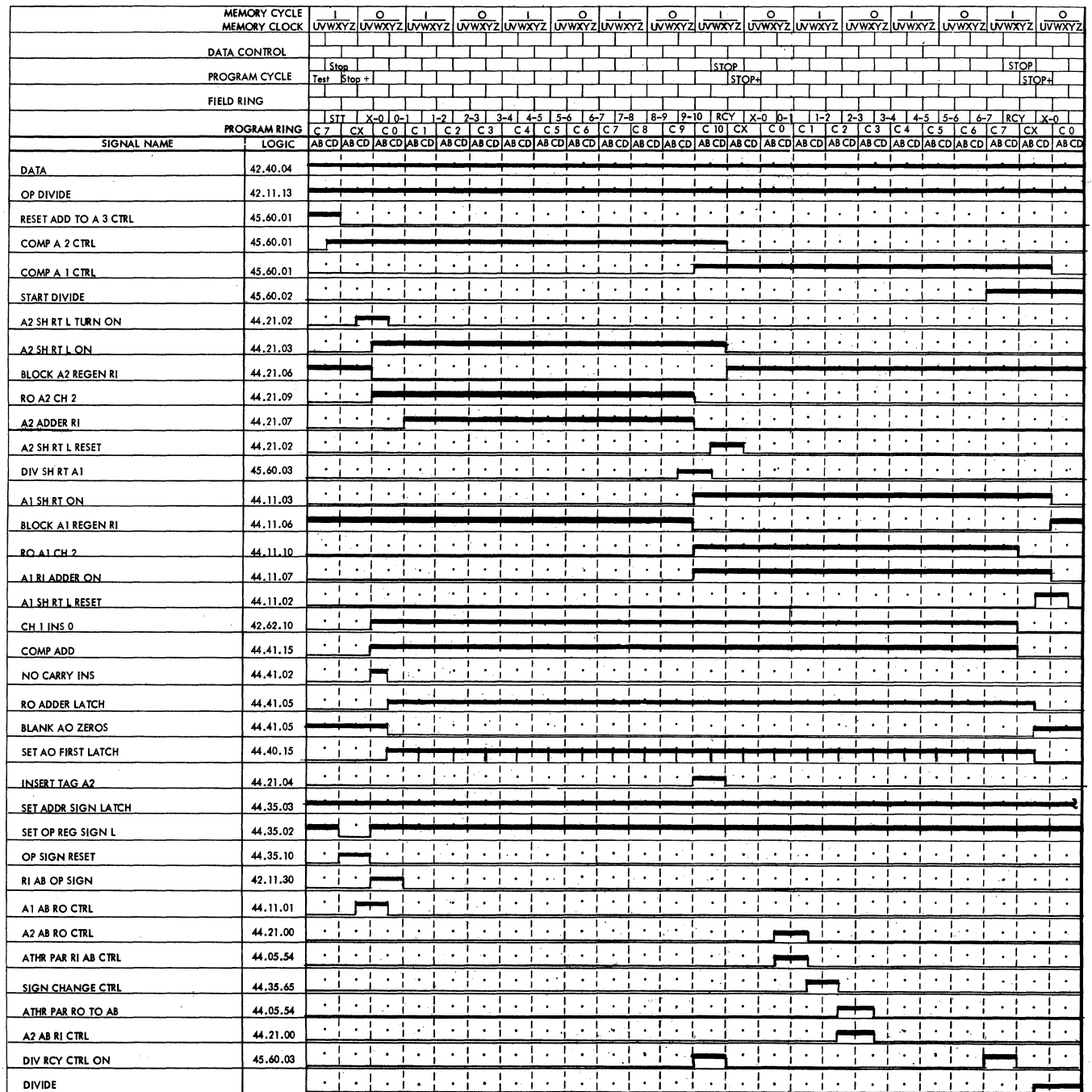


FIGURE 5.3-4. COMPLEMENT A2 AND A1

At C0 of the first cycle, accumulator 2 is serially shifted to the right and read out to channel 1. Zeros are fed to channel 1 from its digit insert. Because the result is to be a 9's complement, the adder entries are gated from the channels with complement-add and no-carry insert. The adder output is serially fed to the high order of accumulator 2 for the first nine digits. On the tenth digit, accumulator 2 reads in a five-bit tag digit. The tenth digit from the adder output is fed into the high order of accumulator 1. In the meantime, accumulator 1 starts to serially shift right to feed channel 2. Channel 1 continues to receive insert zeros. The remaining ten digits from the adder output continue to feed the high order of accumulator 1. The digits shift across accumulator 1 until what had been the high order of accumulator 2 is now in the output latches of accumulator 1. Accumulator 1 now has eleven digits of the dividend and accumulator 2 has nine digits plus the tag used to signal the end of operation.

The remainder of the sign read and adjust operation is performed during the complement cycle. While accumulator 2 is being serially transferred, accumulator 1 is read out in parallel and regenerated. The data is read into the word buffer registers as a convenience. The prime purpose of the transfer is to set the op'd A sign latches with the dividend sign. During the accumulator 1 complement cycle, accumulator 2 is parallel-transferred to the arithmetic register, where the algebraic result of the two factor signs is inserted for the future quotient. If either of the factor signs are alpha, the sign of the quotient is inserted as alpha. The data with the new sign is returned to accumulator 2 before the start of the divide operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Controls		
Comp A2 Ctrl	PC Stop	4E-5E-45.60.01
Comp A1 Ctrl	C10	3G-4G-45.60.01
Divide Rcy Ctrl On (Comp A2)	C10	3G-45.60.03
Prg Cy Stop	C10 CP	42.40.21
Divide Rcy Ctrl On (Comp A1)	C7	4G-45.60.03
Prg Cy Stop	C7 CP	42.40.21
Accumulator 2 to Adder		
A2 Sh Rt L Turn On	CX-0	3D-44.21.02
A2 Sh Rt L On	C0	4D-5D-44.21.03
Block A2 Regen RI	C0	4D-44.21.06
RO A2 Ch 2	C0	4F-5F-44.21.09
A2 Adder RI	C1	3B-44.21.07

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Insert Tag	C10	2F-44.21.04
A2 Sh Rt L Reset	PC Stop	3H-44.21.02
Accumulator 1 to Adder		
Div Sh Rt A1	C9-10	4E-45.60.03
A1 Sh Rt L Turn On	C9-10	4C-44.11.02
A1 Sh Rt On	C10	3E-4E-44.11.03
Block A1 Regen RI	C10	2D-4D-44.11.06
RO A1 Ch 2	C10	4E-5F-44.11.10
A1 RI Adder On	C10	4F-44.11.07
A1 Sh Rt L Reset	CX-0 (cy 3)	3F-44.11.02
Adder Controls		
Ch 1 Ins Zero (On)	C0 (cy 1)	4J-4B-42.62.10
Ch 1 Ins Zero (Off)	CX (cy 3)	5H-4D-42.62.10
Turn On Comp-Add	C0 (cy 1)	3E-4C-44.41.15
Turn Off Comp-Add	CX (cy 3)	44.41.15
No-Carry Insert	C0	3A-4C-44.41.02
RO Adder Latch	C0 CP	4B-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Dividend Sign Set		
Op Sign Reset	CX (cy 1)	44.35.02
A1 RO AB	CX-0	5H-44.11.01
WBR AB RI	CX-0	3C-44.08.12
RI AB Op Sign	C0	42.11.30

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adjust Quotient Sign		
A2 AB RO Ctrl	C0-1 (cy 2)	5D-44.21.00
Athr Par RI AB Ctrl	C0-1	44.05.54
Sign Chg Ctrl	C1-2	6G-44.35.65
Athr Par RO to AB	C2-3	5A-44.05.54
A2 AB RI Ctrl	C2-3	4F-44.21.00

C. Divide Start Cycle (Figure 5.3-5)

The first cycle of the actual divide operation varies slightly from subsequent reductions. First, there has been no overdraw to require the transfer of the word buffer register. Second, the working dividend already has the required eleven digits in accumulator 1 and its output latches. The start divide latch is set at C7 of the complement A1 with the latter latch resetting at C0 of the new cycle. The divide latch is set from the start divide signal during CX-time of the new cycle. At C5-time, the divide left shift add latch is forced to control the shift register operation.

For the reduction cycle, accumulator 1 is serially shifted to the right for read-out. The eleven digits from the register and its output latches are fed to channel 2. Simultaneously, accumulator 3 is serially shifted to the right to read-out. It contains only ten digits, which allows it to be serially regenerated by feeding the output into the high order. The factor is read from the output latches to channel 1. A high-order eleventh digit is supplied with a zero from the channel insert. The adder entries are gated by true-add and no-carry insert. The adder output is serially returned to the high order of accumulator 1. The digits shift across the register until the low order is in the output latches.

During the eleven digit serial transfer of the working dividend (A1) on channel 2, the low-order ten digits are serially inserted into the word buffer register. These digits are right shifted across the register to fill the ten digit positions. The data in this register is used to correct an overdraw condition of the working dividend.

During the adding cycle, accumulator 2 is effectively serial shifted one place to the left. Actually, the register is shifted nine places to the right with serial regeneration. In the process of read-out, all digits are fed to the output latches, but only the last digit to enter is retained. This digit is the high-order remaining digit of the dividend and next in line to be transferred. In the process of regeneration, the last digit is not stored, leaving the high order position blank to insert the first quotient digit. The digit, which was not regenerated, is the same as that in the output latches. Accumulator 2 remains in this position until a divide left shift add cycle.

The shift register is blank at the start of an operation. At C6 of the start divide cycle, a zero is forced into the parallel output latches by the divide left shift add gating.

		MEMORY CYCLE MEMORY CLOCK																
		I		O		I		O		I		O		I		O		
		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		
DATA CONTROL																		
PROGRAM CYCLE		Stop Stop +																
FIELD RING																		
PROGRAM RING		6-7 RCY X-0 0-1 1-2 2-3 3-4 4-5 5-6 6-7 7-8 8-9 9-10 RCY																
SIGNAL NAME		LOGIC		C 6	C 7	CX	C 0	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	CX
		AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD
DATA	42.40.04																	
OP DIVIDE	42.11.13																	
COMP A 1 CTRL	45.60.01																	
START DIVIDE	45.60.02																	
DIVIDE	45.60.02																	
DIVIDE LT SH ADD	45.60.02																	
DIV SH RT A 1	45.60.03																	
A 1 SH RT ON	44.11.03																	
BLOCK A 1 REGEN RI	44.11.06																	
RO A 1 CH 2	44.11.30																	
A 3 SH RT L TURN ON	44.31.02																	
A 3 SH RT ON	44.31.03																	
BLOCK A 3 REGEN RI	44.31.06																	
A 3 SER REGEN	44.31.08																	
RO A 3 CH 1	44.31.09																	
CH 1 INS 0	42.62.10																	
A 1 RI ADDER ON	44.11.07																	
TRUE ADD	44.41.16																	
NO CARRY INS	44.41.03																	
RO ADDER LATCH	44.41.05																	
WBR SH RT TURN ON	44.08.07																	
WBR SER SH RT	44.08.07																	
BLANK WBR REGEN RI	44.08.06																	
WBR RI CH2 CTRL	44.08.27																	
A 2 SH RT L TURN ON	44.21.02																	
A 2 SH RT L ON	44.21.03																	
BLANK A 2 REGEN RI	44.21.06																	
A 2 REGEN RI CTRL	44.21.08																	
SH REG PAR 1 RES	44.50.23																	
ZI ON SHIFT	44.50.23																	
BLOCK SER RO CTRL (PAR RI)	44.50.19																	
SH REG SH LT CTRL	44.50.18																	
DIV RCY CTRL ON	45.60.03																	
CONTINUE DIV ADD	45.60.02																	

FIGURE 5.3-5. START DIVIDE

The output latches are read into the register at C7-time with a left shift. In this case, the bit sets in position 8 and is regenerated until read-out time in the next cycle. In case the first reduction produced an overdraw, the zero in the output latches is ready for insertion into the quotient.

The program ring is recycled at C10-time with CX following in the next digit time. The selection of the type of cycle to follow is dependent on the condition of the adder carry latches at the end of the reduction cycle. If no overdraw has occurred, the no-carry latch is set, which in turn sets the continue divide add latch for the next cycle. When an overdraw does occur, the carry latch is set and the divide left shift add latch is set for the next cycle.

Many of the adding and register control latches are turned off by the divide signal and an appropriate timing pulse. The start divide and the divide left shift add latches are reset at C9. The divide latch remains on until an end divide is indicated.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Controls		
Stt Div	C7	3A-4A-45.60.02
Divide	CX CP	3C-4C-45.60.02
Comp A2 Reset	C0	2H-3H-45.60.01
Divide Lt Sh Add	C5	3E-4E-45.60.02
Div Rcy Ctrl On	C10	3H-45.60.03
Prg Cy Stop	C10 CP	42.40.21
Accumulator 1 to Adder		
Div Sh Rt A1	C0-1	3E-45.60.03
A1 Sh Rt On	C1	3E-4E-44.11.03
RO A1 Ch 2	C0	3D-5F-44.11.10
A1 RI Adder On	C1	4F-44.11.07
A1 Sh Rt L Reset	CX-0 (cy 2)	3F-44.11.02
Accumulator 3 to Adder		
A3 Sh Rt L Turn On	CX-0	4D-44.31.02
A3 Sh Rt On	C0	3D-4D-44.31.03
A3 Ser Regen	C0	3C-4D-44.31.08
RO A3 Ch 1	C0	3D-5B-44.31.09
Ch 1 Ins 0	C10	5B-42.62.10
A3 Sh Rt L Reset	C9-10	4G-44.31.02
Adder Controls		
Turn On True-Add	C0	4D-44.41.16
No-Carry Ins	C0 AP	44.41.03
RO Adder Latch	C0 CP	4B-5B-44.41.05

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Word Buffer Register Read-In		
WBR Sh Rt Turn On	CX-0	4D-44.08.07
WBR Ser Sh Rt	C0	5D-5B-44.08.07
Blank WBR Regen RI	C0	4A-5B-44.08.06
WBR RI Ch 2 Ctrl	C0	4A-44.08.27
Reset WBR Rt Sh	C10	3D-4C-44.08.07
Accumulator 2 Left Shift		
A2 Sh Rt L Turn On	C0-1	44.21.02
A2 Sh Rt L On	C1	4D-5D-44.21.03
A2 Regen RI Ctrl (On)	C1	5F-44.21.08
A2 Regen RI Ctrl (Off)	C9	3H-4G-44.21.08
A2 Sh Rt L Reset	C9-10	3G-44.21.02
Shift Register Read-In		
Sh Reg Par L Reset	C5-6	5A-44.50.23
ZI On Shift	C5-6 AP	44.50.23
Block Ser RO Ctrl (RI)	C6-7	3J-44.50.19
Sh Reg Sh Lt Ctrl	C7	3G-5F-44.50.18

D. Continue Divide Add (Figure 5.3-6)

On the second and subsequent reduction cycles of any quotient digit, the continue divide add latch is set. This results from sensing no-adder carry (no overdraw) in the previous cycle. In the selection of this cycle, we indicate that no-overdraw correction is needed and that the previous reduction should be counted.

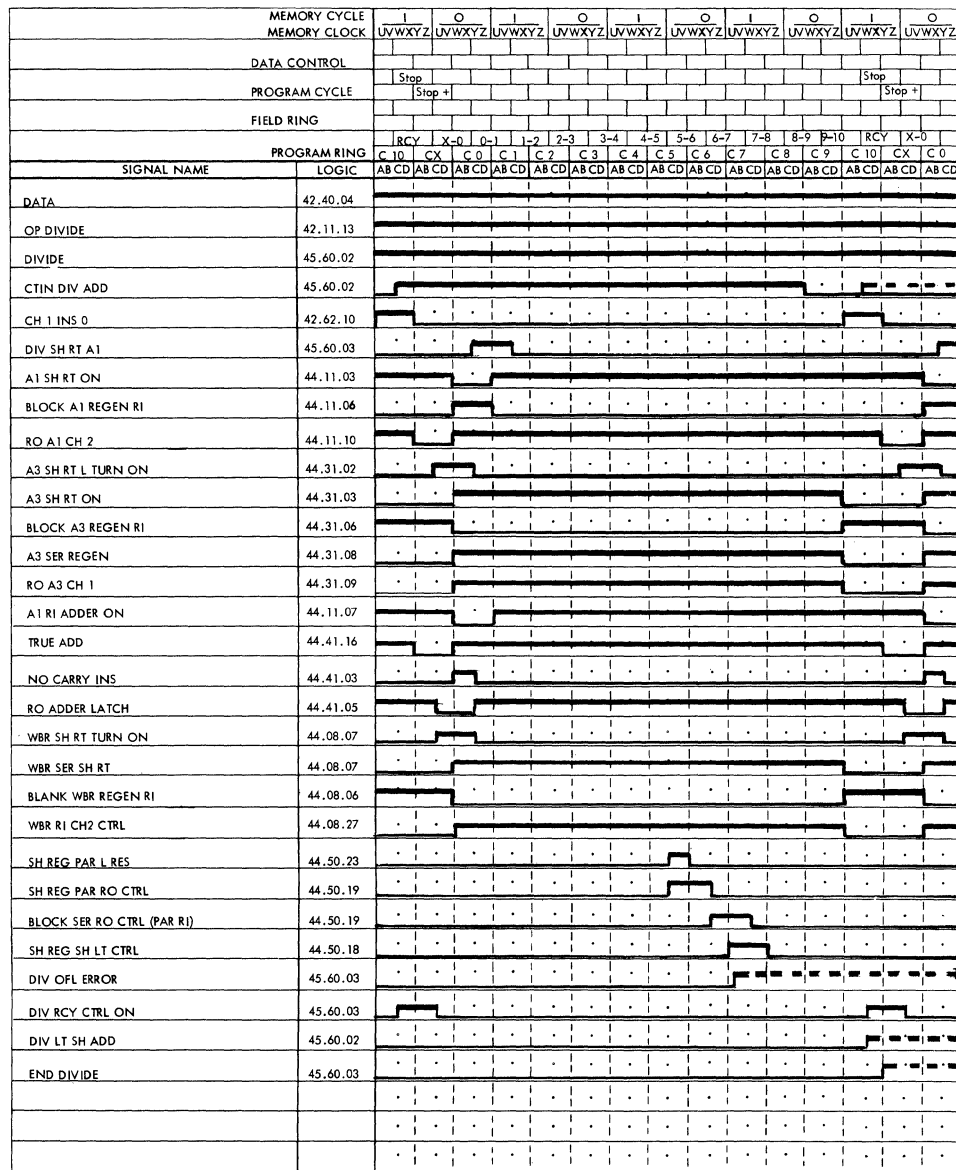


FIGURE 5.3-6. CONTINUE DIVIDE ADD

For the reduction cycle, accumulator 1 is serially shifted to the right for read-out. The eleven digits from the register and its output latches are fed to channel 2. Simultaneously, accumulator 3 is serially shifted to the right to read out to channel 1. The register is serially regenerated by reading the output into the high order. A high-order eleventh digit is supplied as a zero from the channel insert. The adder entries are gated by true-add and no-carry insert. The adder output is serially returned to the high order of accumulator 1. The digits shift across the register until the low order is in the output latches.

During the eleven digit serial transfer of the working dividend (A1) on channel 2, the low order ten digits are serially inserted into the word buffer register. These digits are right shifted across the register to fill the ten digit positions. The data in this register is used to correct an overdraw condition of the working dividend.

Accumulator 2 is not operated during a continue divide add cycle. The effective left shift is made only during a start divide or divide left shift add cycle, which represents the first reduction of each quotient digit.

The shift register contains the one-upped or true reduction count in the core register. At C6-time, the parallel output latches are cleared and the core value entered. The output latches are then read back into the register at C7-time with a left shift. The bit sets in a position one digit greater than it read out. It is regenerated in this position until the next read-out time. The output latches retain the previous digit ready for insertion into the quotient.

The program ring is recycled at C10-time with CX following in the next digit time. Either another continue divide add cycle or a divide left shift add cycle may follow depending on the adder carry latches at the end of the reduction. Many of the control latches are turned off by the divide signal at the end of the cycle, because continue divide add resets at C9-time.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Controls		
Ctin Div Add	PC Stop CP	3G-45.60.02
Div Rcy Ctrl On	C10	3H-45.60.03
Prg Cy Stop	C10 CP	42.40.21
Accumulator 1 to Adder		
Div Sh Rt A1	C0-1	3E-45.60.03
A1 Sh Rt On	C1	3E-4E-44.11.03
RO A1 Ch 2	C0	3E-5F-44.11.10
A1 RI Adder On	C1	4F-44.11.07
A1 Sh Rt L Reset	CX-0 (cy 2)	3F-44.11.02
Accumulator 3 to Adder		
A3 Sh Rt L Turn On	CX-0	4D-44.31.02
A3 Sh Rt On	C0	3D-4D-44.31.03
A3 Ser Regen	C0	3C-4D-44.31.08
RO A3 Ch 1	C0	3D-5B-44.31.09
Ch 1 Ins 0	C10	5B-42.62.10
A3 Sh Rt L Reset	C9-10	4G-44.31.02

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder Controls		
Turn On True-Add	C0	4D-44.41.16
No-Carry Ins	C0 AP	44.41.03
RO Adder Latch	C0 CP	4B-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Word Buffer Register Read-In		
WBR Sh Rt Turn On	CX-0	4D-44.08.07
WBR Ser Sh Rt	C0	5D-5B-44.08.07
Blank WBR Regen RI	C0	4A-5B-44.08.06
WBR RI Ch 2 Ctrl	C0	4A-44.08.27
Reset WBR Rt Sh	C10	3D-4C-44.08.07
Shift Register Controls		
Sh Reg Par L Reset	C5-6	5A-44.50.23
Sh Reg Par RO Ctrl	C5-6 AP	44.50.19
Block Ser RO Ctrl (RI)	C6-7	3J-44.50.19
Sh Reg Sh Lt Ctrl	C7	3G-5F-44.50.18

E. Divide Left Shift Add (Figure 5.3-7)

When the previous reduction cycle results in an adder carry output, it indicates that the effective sign has been reversed. The working dividend factor is no longer a complement value. This condition results in the setting of the divide left shift add latch. In addition to the normal reduction, the latch provides for overdraw correction, an effective left shift of the dividend, and the insertion of the developed quotient digit.

During each divide add or reduction cycle, the ten low-order digits of the working dividend are serially entered into the word buffer register. The new value always replaces the previous entry. When an overdraw occurs, the high-order digit, which was dropped, is always a zero. The word buffer register, therefore, contains the true value of the working dividend prior to the last reduction. If this value is parallel-transferred back to accumulator 1 in the normal manner, two things are accomplished. First, the overdraw correction is effected in one digit time. Second, the position of entry is such that it produces a left shift of the factor. The new eleventh digit for the low-order position was previously shifted to the output latches of accumulator 2.

During the eleven digit serial transfer of the working dividend (A1) on channel 2, the low order ten digits are serially inserted into the word buffer register. These digits right shift across the register to fill the ten digit positions. The data in this register is used to correct an overdraw condition if it occurs as a result of this reduction.

During the adding cycle, accumulator 2 is again right shifted to effect a left shift and to read out the next high order dividend digit. The digit previously left in the output latches has entered the problem. The vacated position of the high order now contains a quotient digit. The register serially shifted nine positions to the right, leaving the next dividend digit in the output latches. Only the first eight digits are regenerated at the high order leaving a blank spot for the next quotient digit.

The shift register is not read out to the output latches on a divide left shift add cycle. At C6, the register is blanked and a zero is forced into the output latches. The output latches are read into the register at C7-time, with a left shift. The bit again sets in the one position and is regenerated until the next read-out.

The program ring is recycled at C10-time with CX following in the next digit time. The adder carry is again used to control if another divide left shift add cycle or a continue divide add cycle is to follow. When two divide left shift add cycles occur in a row, a quotient digit of zero has been developed. Many of the control latches are turned off by the divide signal at the end of the cycle, because divide left shift add resets at C9-time.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Controls		
Div Lt Sh Add	PC Stop CP	2E-4E-45.60.02
Div Rey Ctrl On	C10	3H-45.60.03
Prg Cy Stop	C10 CP	42.40.21
Overdraw Correction		
WBR AB RO	CX-0	3E-44.08.12
A1 AB RI Ctrl	CX-0	3G-44.11.00
Quotient Digit Insert	C0 CP	3F-4A-44.20.20
Accumulator 1 to Adder		
Div Sh Rt A1	C0-1	3E-45.60.03
A1 Sh Rt On	C1	3E-4E-44.11.03
RO A2 Ch 2	C0	5D-44.21.09
RO A1 Ch 2	C1	2E-5F-44.11.10
A1 RI Adder On	C1	4F-44.11.07

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A1 Sh Rt L Reset	CX-0 (cy 2)	3F-44.11.02
Accumulator 3 to Adder		
A3 Sh Rt L Turn On	CX-0	4D-44.31.02
A3 Sh Rt On	C0	3D-4D-44.31.03
A3 Ser Regen	C0	3C-4D-44.31.08
RO A3 Ch 1	C0	3D-5B-44.31.09
Ch 1 Ins 0	C10	5B-42.62.10
A3 Sh Rt L Reset	C9-10	4G-44.31.02
Adder Controls		
Turn On True-Add	C0	4D-44.41.16
No-Carry Ins	C0 AP	44.41.03
RO Adder Latch	C0 CP	4B-5B-44.41.05
Blank AO Zeros	C0 CP	5D-44.41.05
Set AO First Latches	C0 CP	44.40.15
Word Buffer Register Read-In		
WBR Sh Rt Turn On	CX-0	4D-44.08.07
WBR Ser Sh Rt	C0	5D-5B-44.08.07
Blank WBR Regen RI	C0	4A-5B-44.08.06
WBR RI Ch 2 Ctrl	C0	4A-44.08.27
Reset WBR Rt Sh	C10	3D-4C-44.08.07
Shift Register Controls		
Sh Reg Par L Reset	C5-6	5A-44.50.23
Ser RO Ctrl Reset	C5-6	3J-44.50.19
ZI On Shift	C5-6	44.50.23
Block Ser RO Ctrl (RI)	C6-7	3J-44.50.19
Sh Reg Sh Lt Ctrl	C7	3G-5F-44.50.18

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Accumulator 2 Left Shift		
A2 Sh Rt L Turn On	C0-1	3B-44, 21.02
A2 Sh Rt L On	C1	4D-5D-44, 21.03
A2 Regen RI Ctrl (On)	C1	3E-5F-44, 21.08
A2 Regen RI Ctrl (Off)	C9	3H-4G-44, 21.08
A2 Sh Rt L Reset	C9-10	3G-44, 21.02

F. End Divide - Complement Remainder (Figure 5.3-8)

The divide operation continues until ten quotient digits have been developed. The individual digits each require one divide left shift add cycle and continue divide add cycles equal to the value of the digit. An all zero quotient requires only the ten divide left shift add cycles, while all nines requires in addition ninety continue divide add cycles. If the factors are such that the divisor is not greater than the accumulator 1 portion of the dividend at the start of the operation, the overdraw point can be reached in nine reduction cycles. The tenth reduction causes the shift register bit to shift into the A first output latch to indicate an error.

The five-bit tag digit read into accumulator 2 at the start of the operation shifts each time the register is cycled. On the tenth shift, it remains in the output latches to signify the last digit has been used. When the overdraw is reached and the divide left shift add latch is set, the tag is gated to set the end divide latch. A final cycle is required to complete the various phases of the operation.

The presence of the divide left shift add signal in the end divide cycle allows the use of several normal gates. The working dividend, which now becomes the remainder, has been overdrawn. The value in the word buffer register is parallel-transferred to accumulator 1. In the process, it is left shifted placing the remainder fully within the register.

The divide left shift add signal also completes the insert and alignment of the quotient. The last quotient digit from the shift register output latches is inserted in the high order blank position of accumulator 2. The factor is still positioned one digit to the right with the low order in position 0 of the register. The nine-digit right shift with serial regeneration is used to effect the shift. The regeneration takes place for the full nine digits, instead of eight as in previous cycles. This results from the use of the end divide signal with a later digit gate, instead of the divide signal.

The end divide signal causes accumulator 1 to serial shift to the right and read out to channel 2, to allow recomplementing the remainder. Accumulator 3 is not read out, because the divide signal is now down. Channel 1 is fed with zeros from the channel insert. The adder entries are gated with complement-add and no-carry insert to produce a 9's complement. The adder output is serially returned to the high order of accumulator 1. The digits right shift until the low order is in position 9 of the register.

The program ring is stopped at C11 by forcing the program cycle ring at the last + position. The divide left shift add latch is reset at the usual C9-time. The end divide latch is held until the fall of the op divide signal after the start of the next instruction control. The end data op signal comes up for C10-11 to turn off the data latch.

At the end of the operation the dividend remainder is in accumulator 1. The quotient is in accumulator 2 and the original divisor is in accumulator 3. The original dividend is no longer available.

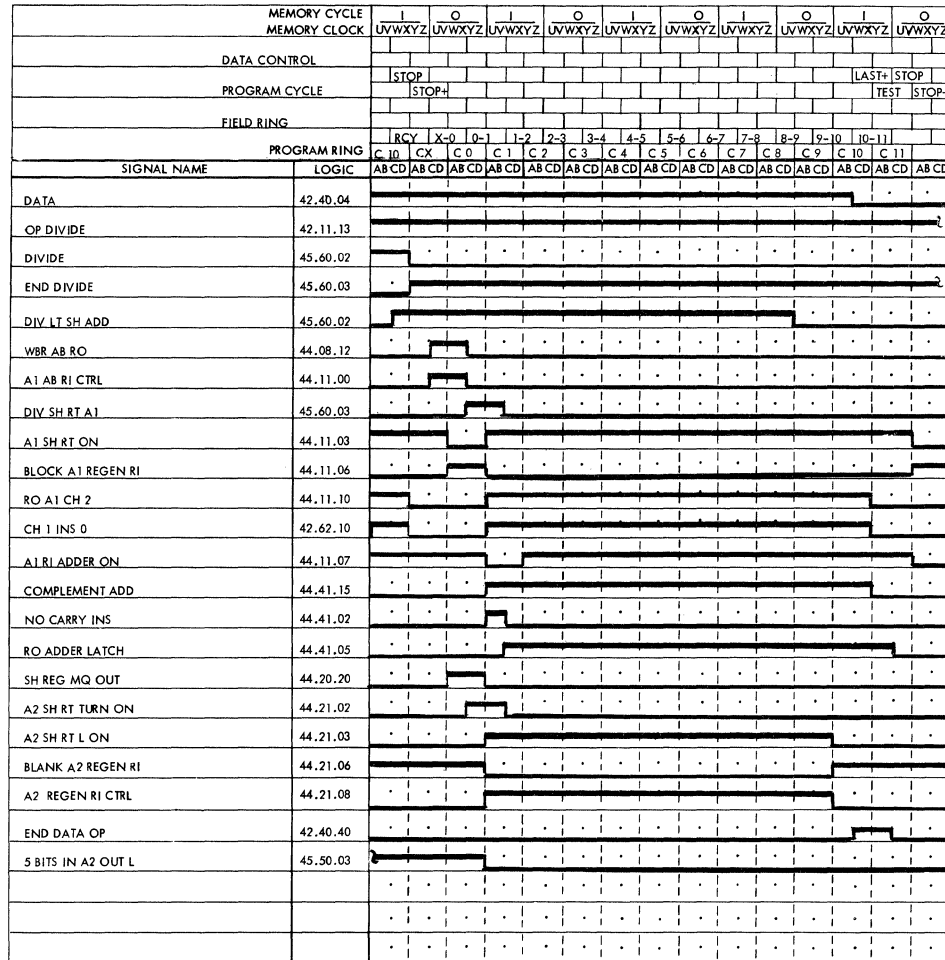


FIGURE 5.3-8. DIVIDE END

Major Signals	Key Timing	Logic and Location
Operation Controls		
5 Bits in A2 Out L	C9	2F-45.50.03
Div Lt Sh Add	PC Stop CP	2E-4E-45.60.02
End Divide	CX-0	3A-4A-45.60.03

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Divide L Reset	CX-0	3D-45.60.02
Set Prg Cy Last +	C10 CP	
Set End Data Op	C10 CP	4F-42.40.40
Overdraw Correction		
WBR AB RO	CX-0	3E-44.08.12
A1 AB RI Ctrl	CX-0	3G-44.11.00
Quotient Digit Insert	C0 CP	3F-4A-44.20.20
Accumulator 2 Left Shift		
A2 Sh Rt L Turn On	C0-1	3B-44.21.02
A2 Sh Rt L On	C1	4D-5D-44.21.03
A2 Regen RI Ctrl (On)	C1	3E-5F-44.21.08
A2 Regen RI Ctrl (Off)	C10	3J-4G-44.21.08
A2 Sh Rt L Reset	C9-10	2H-44.21.02
Accumulator 1 to Adder		
Div Sh Rt A1	C0-1	3F-45.60.03
A1 Sh Rt On	C1	3E-4E-44.11.03
RO A1 Ch 2	C0-1	2E-5F-44.11.10
A1 RI Adder On	C1	4F-44.11.07
A1 Sh Rt L Reset	PC Stop +	4H-44.11.02
Adder Controls		
Ch 1 Insert 0	C1	4H-4B-42.62.10
Turn On Comp-Add	C1	2D-4C-44.41.15
No-Carry Ins	C1 AP	44.41.02
RO Adder Latch	C1 CP	4B-5B-44.41.05
Blank AO Zeros	C1 CP	5D-44.41.05
Set AO First Latches	C1 CP	44.40.15

FIGURE 5.3-9. DIVISION PROBLEM

	SR	OL	WBR	A 1	OL	A 2	OL	RMKS
START	0	X	XXXXXXXXXX	0000089592		3806519862		
COMP ADD A 1 & 2				9999910407	6	* 193480137	0	TAG INS
START DIVIDE	1	0	9999104076	C 0035603157	7	93480137*	1	O'DRAW
DIV LEFT SHIFT				9999104076	7	0 93480137*	1	
ADD	1	0	9991040761	C 0034796826	2	3480137*0	9	O'DRAW
DIV LEFT SHIFT				9991040761	2	0 3480137*0	9	
ADD	1	0	9910407619	C 0026733512	0	480137*00	3	O'DRAW
DIV LEFT SHIFT				9910407619	0	0 480137*00	3	
ADD	1	0	9104076193	N 9946100369	4	80137*000	4	
CTIN DIV ADD	2	1	9461003694	N 9981793119	5			
CTIN DIV ADD	3	2	9817931195	C 0017485869	6			O'DRAW
DIV LEFT SHIFT				9817931195	6	2 80137*000	4	
ADD	1	0	8179311954	N 9853623945	5	0137*0002	8	
CTIN DIV ADD	2	1	8536239455	N 9889316695	6			
CTIN DIV ADD	3	2	8893166956	N 9925009445	7			
CTIN DIV ADD	4	3	9250094457	N 9960702195	8			
CTIN DIV ADD	5	4	9607021958	N 9996394945	9			
CTIN DIV ADD	6	5	9963949459	C 0002087696	0			O'DRAW
DIV LEFT SHIFT				9963949459	0	5 0137*0002	8	
ADD	1	0	9639494598	N 9999642209	9	137*00025	0	
CTIN DIV ADD	2	1	9996422099	C 0035334960	0			O'DRAW
DIV LEFT SHIFT				9996422099	0	1 137*00025	0	
ADD	1	0	9964220990	C 0032114849	1	37*000251	1	O'DRAW
DIV LEFT SHIFT				9964220990	1	0 37*000251	1	
ADD	1	0	9642209901	N 9999913740	2	7*0002510	3	
CTIN DIV ADD	2	1	9999137402	C 0035606490	3			O'DRAW
DIV LEFT SHIFT				9999137402	3	1 7*0002510	3	
ADD	1	0	9991374023	C 0034830152	4	*00025101	7	O'DRAW
DIV LEFT SHIFT				9991374023	4	0 *00025101	7	
ADD	1	0	9913740237	C 0027066773	8	000251010	*	O'DRAW & TAG RO
DIV LEFT SHIFT				9913740237	8	0 000251010	*	
COMP REM				0086259762	0	0 002510100	0	END OP
A 3			REMAINDER		QUOTIENT			
0356927501								
DIVISOR			X = UNUSED FACTOR		X = INSERT DIGIT			

QUOTIENT DEVELOPMENT DURING DIVISION

DIVIDEND	00000895923806519862
DIVISOR	0356927501
QUOTIENT	0002510100
REMAINDER	0086259762

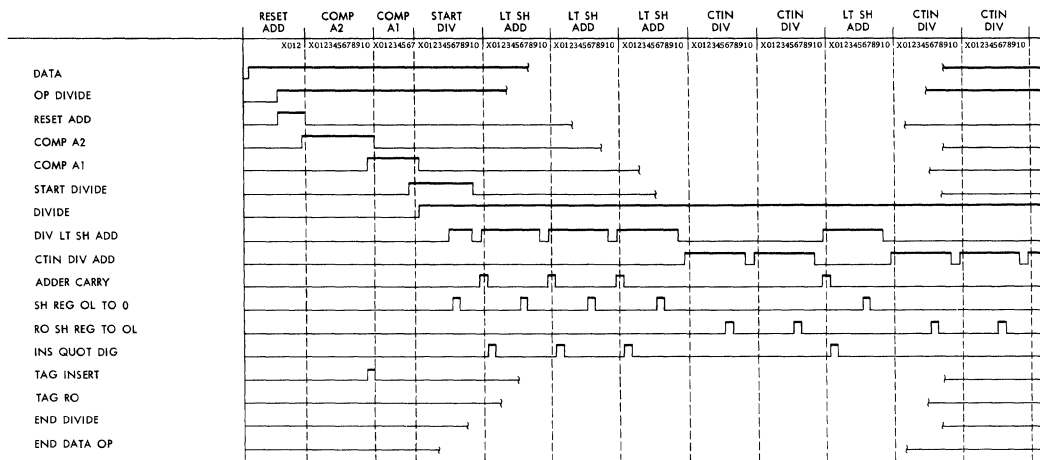


FIGURE 5.3-10A. TIMING CHART FOR DIVISION PROBLEM

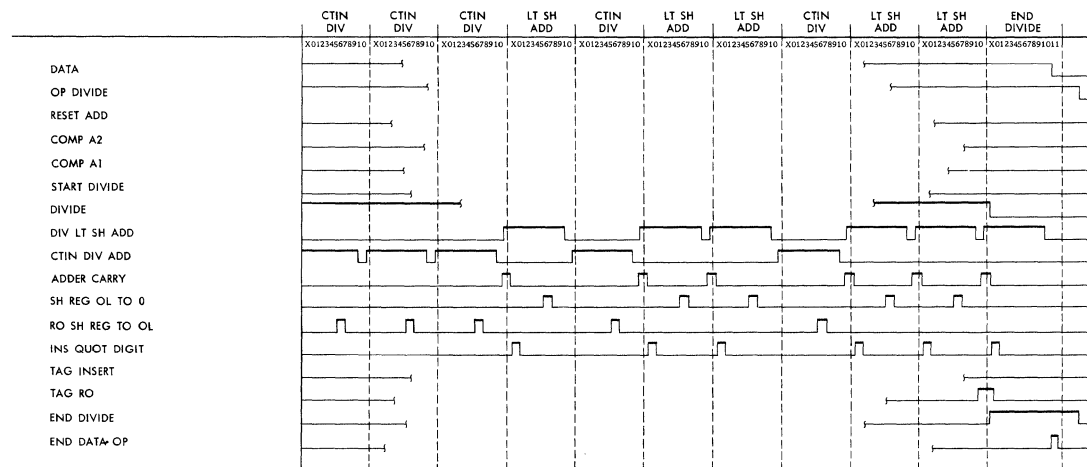


FIGURE 5.3-10B. TIMING CHART FOR DIVISION PROBLEM

5.4.00 SHIFT OPERATIONS

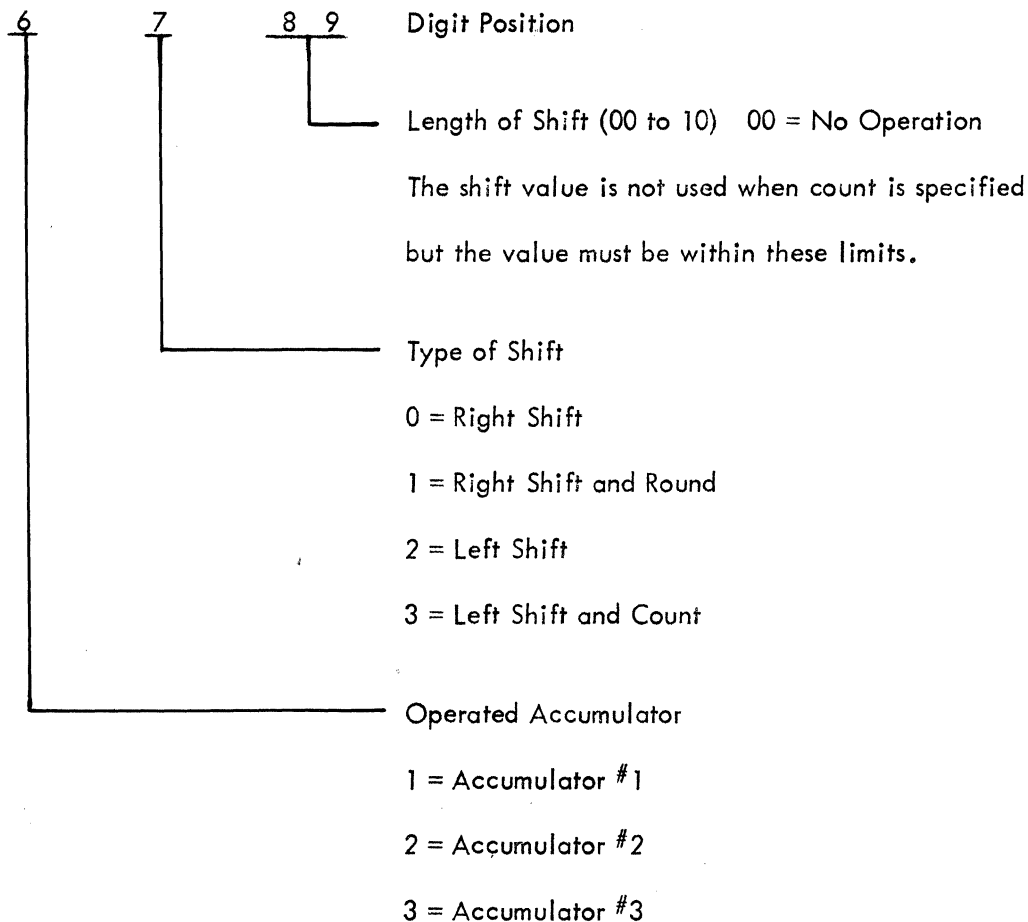
The shift function is provided to allow positioning factors for arithmetic or control operations. The basic shift operations are divided into two groups. One is used to process a single word-factor, which may be located in any accumulator. The second is used to process double-word factors like the product of a multiplication or a dividend for division. Accumulators 1 and 2 are used with accumulator 1 as high order. The factors are placed in the accumulators on a previous operation and the result remains in the same area at the end of the shift operation. Vacated areas of the accumulators are filled with zeros while the digits shifted out are lost.

A special group of double-word shift codes allow splitting the factor at a specified point. One portion is moved by the indicated shift value. Zeros are inserted between the two groups. All parallel transfers are made on the arithmetic bus.

SINGLE SHIFT

The basic operation code of +50 is augmented to indicate the accumulator, the type of shift, and the length of the shift. The address portion of the instruction word does not specify a data word. The address may be indexed in the normal manner to adjust the instructions prior to the shift operation. The field control portion of the instruction word is only used with a shift and count operation. In this operation, it specifies the address of an index word to be used to store the count value.

The address portion is broken down as follows:



Shift Left or Right (Figure 5.4-1)

A single shift either to the right or to the left follows the same basic pattern. The units position of the data address is read into the shift register to control the shifting operations. Zeros are entered to fill the portion of the register being vacated at either the left or right side. Any significant digits, which may be shifted out at the end of the register, are lost.

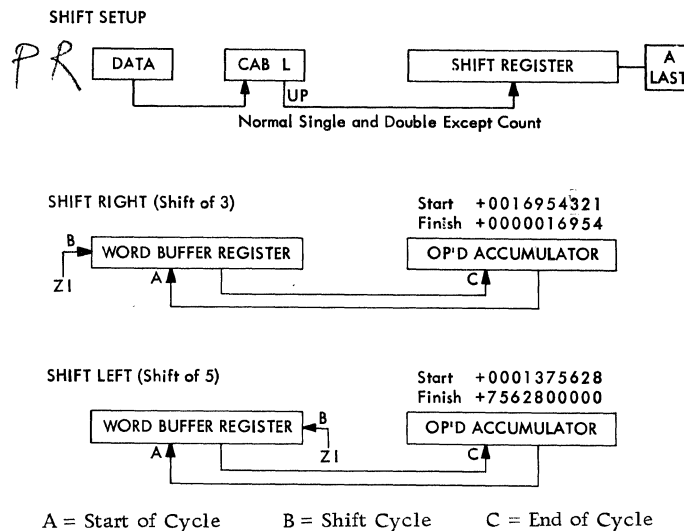


FIGURE 5.4-1. SINGLE SHIFT RIGHT OR LEFT

At the start of the data cycle, the program register D is read out to the computer address bus and regenerated. The address is used to set the computer address bus latches which serve as part of the address matrix. The units position value is read out to set the shift register for the shift count. If the units position is zero and the tens position is a one, the shift value is inserted at the high order of the shift register to give a count of ten.

The word buffer register is used as the working register for both right and left shifts. The operated accumulator is read out in parallel to the arithmetic bus and regenerated. The data reads into the word buffer register. The shift register is serially right shifted until it reaches the low order A-last latch. Simultaneously, the word buffer register is serially shifted either right or left as indicated. Zeros are entered at the left or right end, respectively. At the completion of the shift, the data is read out of the word buffer register via the arithmetic bus to the operated accumulator to replace the original value.

Shift Right and Round

A rounding operation is applied to the basic right shift when specified. If the value of significant digits being dropped is greater than five tenths, a one is added to the low-order position. In operation, a five is added to the last digit position dropped and the carry added to the retained value.

The setup and shift operations of a right shift and round are identical to the basic right shift. The shift register is set from the computer address bus latches. The operated accumulator is parallel-transferred to the word buffer register. During the right shift, the positions being shifted out pass through the output latches. The clearing of the output latches is blocked to hold an eleventh digit, which is the high order of those dropped.

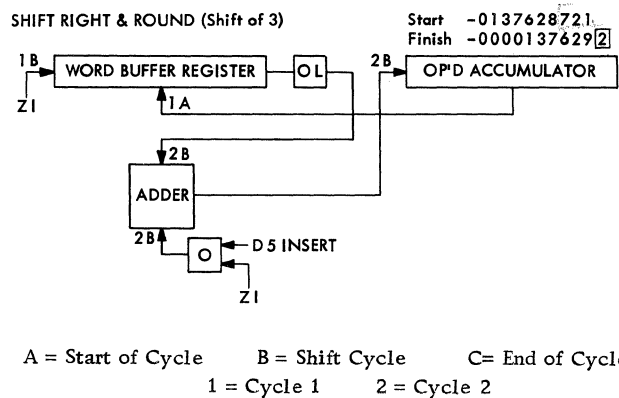


FIGURE 5.4-2. SHIFT RIGHT AND ROUND

Rounding Operation (Figure 5.4-2). A second cycle is initiated to perform the rounding operation. The eleven digits of the word buffer register and its output latches are read serially to channel 1 and adder entry A. Adder entry B with five-add control is fed from the channel 2 digit insert. A five is inserted on the first digit time opposite the digit from the output latches. If the digit is five or greater, a carry is developed to add to the low order register position. Zeros are inserted on channel 2 for the remaining ten digits. The adder output is serially returned to the operated accumulator to replace the original factor. The eleven digit cycle causes the tested digit to shift out of the accumulator leaving the rounded digit in the low order position.

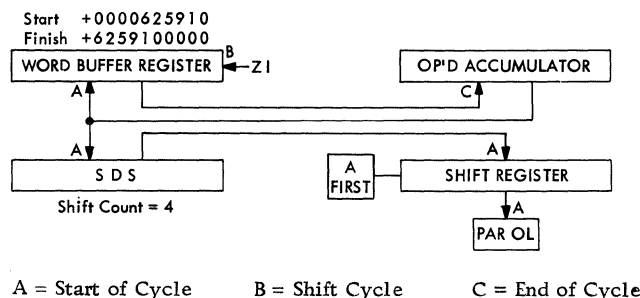


FIGURE 5.4-3. SHIFT LEFT AND COUNT (CYCLE 1)

Shift Left and Count (Figure 5.4-3)

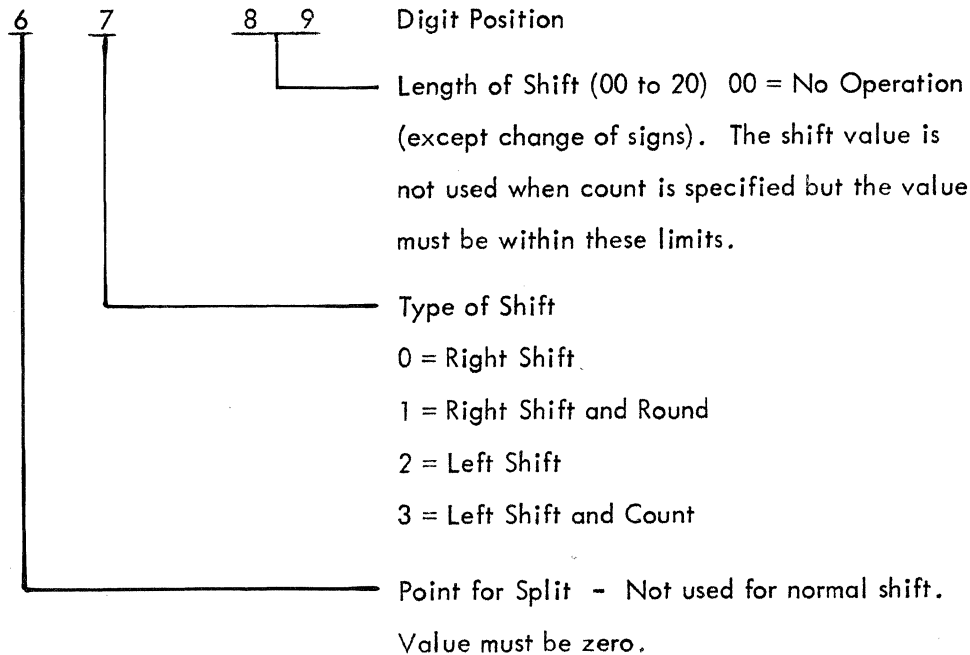
The shift left and count operation differs from the basic left shift in the setup control. In a left shift with a predetermined count, digits may be lost from the high order end. The shift and count operation first determines the number of high order zeros and then proceeds to shift by that value. The shift value is recorded for future use in the index word location specified by the control portion of the instruction word. The shift cycle is similar to the basic left shift.

The operated accumulator is parallel-transferred to the word buffer register. During the transfer, the data is sampled by the significant digit scanner. The scanner locates the high order significant digit and sets the shift register. Before starting the shift, the shift register is read out to set its parallel output latches and is read back into the register for use. The shift register is serially shifted to the left to count the zeros because it was set for the significant digits. The shift is continued until it reaches the high order and sets the A-first latch. The word buffer register is shifted left simultaneously. At the end of the shift cycle, the word buffer register is parallel-transferred to the operated accumulator to replace the original factor.

The field control portion of the instruction word is only used with a shift and count operation. In this operation, it specifies the address of an index word to be used to store the correct value.

In a double shift operation, the signs of two words are involved. On a right coupled shift the sign of accumulator 2 is set to that of accumulator 1. On a left coupled shift, the sign of accumulator 1 is set to that of accumulator 2. These sign changes are made even for a shift of zero places.

The address portion for a normal double shift is broken down as follows:



Shift Right

In a double shift right operation, the combined factors in accumulators 1 and 2 are shifted right by the shift value indicated in the instruction. The low order positions of the accumulator 2 factor are shifted out and are lost. The low order positions of the accumulator 1 factor are entered into the high order of accumulator 2. Zeros are entered into the high order position of accumulator 1 to fill the positions left vacant.

The computer address bus latches are set from program register D and the register is regenerated. The shift register is set from the units position latches in the same manner as for a single shift. The tens position latches are used to set controls to effect a shift of ten by parallel transfers between registers. A shift value of twenty sets the controls for greater than ten and sets the shift register in the high order for a shift of ten. The shift operation takes but one program cycle in all cases.

Shift of Less Than Ten (Figure 5.4-5). Accumulator 2 is read out in parallel to the arithmetic bus and regenerated. The data is read into the word buffer register, which is functionally connected to serve as the low order to Accumulator 1. The shift register

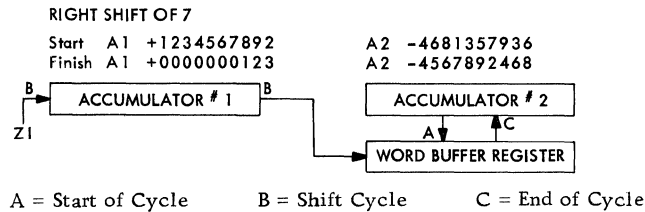


FIGURE 5.4-5. DOUBLE SHIFT RIGHT--LESS THAN TEN

is serially right shifted until it reads out at the low order to set the A last latch. Simultaneously, accumulator 1 and the word buffer register are right shifted a like number of positions. Digits reading out of position 9 of accumulator 1 are read into position 0 of the word buffer register. Zeros are inserted at position 0 of accumulator 1. At the completion of the shift, the data is read out of the word buffer register in parallel to the arithmetic bus. It is read into accumulator 2 to replace the original value.

Shift of Ten or Greater (Figure 5.4-6). At the start of the operation, zeros are inserted in all positions of the arithmetic register with its zero insert latches. Accumulator 1 is read out in parallel to the arithmetic bus and regenerated. The data is read into the word buffer register, which now functions independently, similar to a single shift. The shift register and the word buffer register are serially shifted as for the less than ten operation except that accumulator 1 is not used. Zeros are inserted at position 0 of the word buffer register. At the completion of the shift, the data is read out in parallel from the word buffer register to the arithmetic bus. It is read into accumulator 2 to replace the original value. The zeros in the arithmetic register are read out in parallel to the arithmetic bus and into accumulator 1 to replace the original value. On a right shift of ten or greater, the high-order factor word is completely transferred out of accumulator 1 and the low-order factor word is completely shifted out and lost.

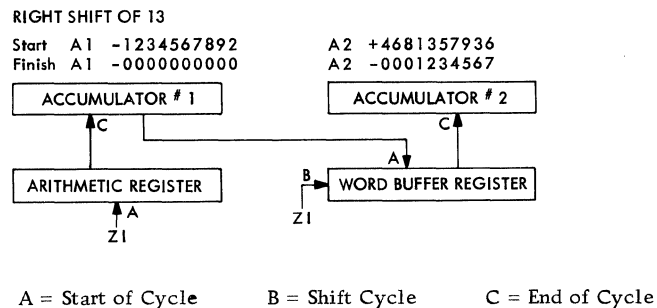


FIGURE 5.4-6. DOUBLE SHIFT RIGHT--TEN OR GREATER

Shift Left

In a double shift left operation, the combined factors of accumulators 1 and 2 are shifted left by the shift value. The high-order positions of the accumulator 1 factor are shifted out and are lost. The high-order positions of the accumulator 2 factor are entered into the low order of Accumulator 1. Zeros are inserted into the low-order positions of Accumulator 2 to fill the positions left vacant.

The setup of the shift register and the operating controls from the computer address bus latches is identical to that of shift right.

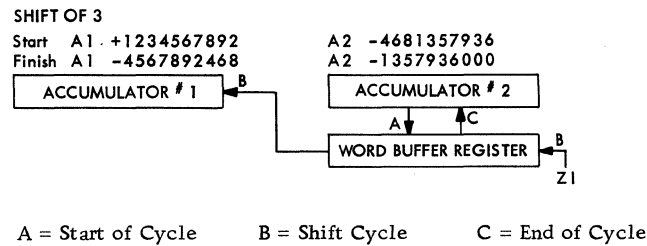


FIGURE 5.4-7. DOUBLE SHIFT LEFT--LESS THAN TEN

Shift of Less Than Ten (Figure 5.4-7). Accumulator 2 is read out in parallel to the arithmetic bus and regenerated. The data is read into the word buffer register, which is functionally connected to serve as the low order to accumulator 1. The shift register is serially right shifted until it reads out of the low order to the A last latch. Simultaneously, accumulator 1 and the word buffer register are left shifted a like number of positions. Digits reading out of position 0 of the word buffer register are read into position 9 of accumulator 1. Zeros are inserted at position 9 of the word buffer register. At the completion of the shift, the data is read out in parallel from the word buffer register to the arithmetic bus. It is read into accumulator 2 to replace the original value.

Shift of Ten or Greater (Figure 5.4-8). At the start of the operation, zeros are inserted in all positions of the arithmetic register with its zero insert latches. Accumulator 2 is transferred in parallel to the word buffer register as for a shift of less than ten. The shift register is right shifted to A last and the word buffer register is left shifted simultaneously. The word buffer register operates independently and accumulator 1 is not shifted. Zeros are inserted at position 9 of the word buffer register. At the completion of the shift, the data is read out in parallel from the word buffer register to the arithmetic bus. It is read into accumulator 1 to replace the original value. The zeros in the arithmetic register are read out in parallel to the arithmetic bus and into accumulator 2 to replace the original value. On a left shift of ten or greater, the low-order factor word is completely transferred out of accumulator 2 and the high order factor word is completely shifted out and lost.

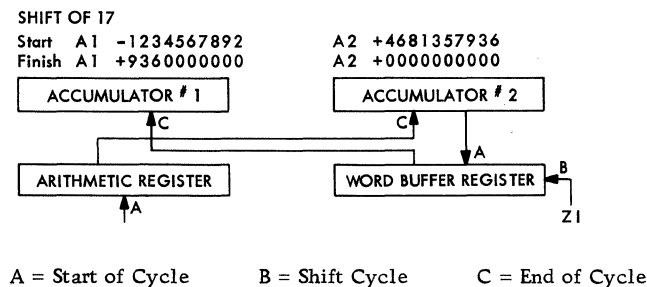


FIGURE 5.4-8. DOUBLE SHIFT LEFT--TEN OR GREATER

Shift Right and Round

The double shift right and round operation is similar to the single-word operation, the object being to adjust the factor, after shift, to the nearest whole digit. To satisfy the extremes, it is necessary to operate the adder for the full capacity of both registers or twenty one digits. This requires two additional program cycles.

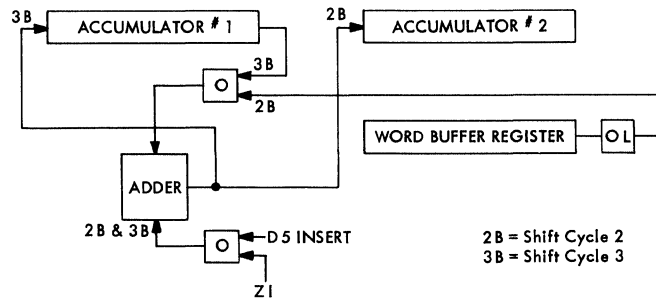


FIGURE 5.4-9. ROUND CYCLE---DOUBLE RIGHT SHIFT

The setup of the computer address bus latches and the shift register are identical to that for a basic shift operation. The shift follows the same pattern as the respective shift right operation except that the word buffer register is not parallel transferred to accumulator 2. At the end of the right shift of the word buffer register, the last digit read out is held in the output latches.

Rounding Operation (Figure 5.4-9). A second cycle is initiated to start the rounding operation. The eleven digits of the word buffer register and its output latches are read serially to channel 1 and adder entry A. Adder entry B with true-add control is fed from the channel 2 digit insert. A five is inserted on the first digit time and zeros for the remaining ten digits. The adder output is serially returned to accumulator 2 to replace the original factor. The eleven digit cycle causes the tested digit to shift out of the accumulator leaving the rounded digit in the low order position. A third cycle is used to serially read out accumulator 1 to channel 1 and adder entry A to accommodate a carry from the low-order factor word. Zeros are inserted on channel 2 as in the second cycle. The adder output serially reads into accumulator 1 to end the operation.

Shift Left and Count

The double shift left and count operation is similar to the single-word operation. The shift value is determined by sensing the factor with the significant digit scanner. The value is set in the shift register for shift count and in its parallel output latches for storage. Like the single shift operation, the count is stored in the specified index word.

Accumulator 1 Non-Zero (Figure 5.4-10). Accumulator 1 is read out in parallel to the arithmetic bus and regenerated. The data is read into the word buffer register and is sampled by the significant digit scanner. With the sensing of a significant digit, the shift register is set for the shift count. Accumulator 2 is then read out in parallel to the arithmetic bus and read into accumulator 1. Accumulator 1 functions as the low order and the word buffer register the high order. Digits reading out of position 0 of accumulator 1 are read into position 9 of the word buffer register. Zeros are inserted at position 9 of accumulator 1. The shift register is read out to its parallel output latches and regenerated by reading back in. The shift register is serially left shifted until it reads out to the A-first latch. Simultaneously, the word buffer register and accumulator 1 are left shifted. At the completion of the shift, accumulator 1 is parallel-transferred back to accumulator 2 to replace its original factor. The word buffer register is parallel-transferred back to accumulator 1 to replace the working data it contained.

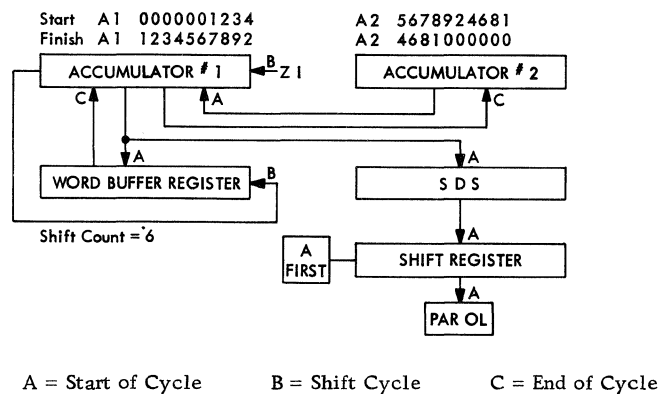


FIGURE 5.4-10. DOUBLE SHIFT LEFT AND COUNT--A1 NON ZERO

Accumulator 1 Zero (Figure 5.4-11). If the significant digit scanner senses all zeros during the transfer of accumulator 1 to the word buffer register, the shift register is not set. The ten's count latches are set to one. When accumulator 2 is read out to the arithmetic bus, it is read into the word buffer register to replace the accumulator 1 factor. The significant digit scanner senses this factor during the transfer to set the shift register and its parallel output latches. The shift register is left shifted until it reads out to the A-first latch. Only the word buffer register is left shifted for the count and zeros are inserted into position 9. At the completion of the shift, accumulator 1 parallel transfers its zeros to accumulator 2. The word buffer register parallel transfers the resulting shift factor to accumulator 1 to replace the zeros.

If both accumulators were zero, the shift register parallel output latches are set to zero and the ten's count latches are set to two for a count of twenty. No actual shift takes place and both accumulators remain at zero.

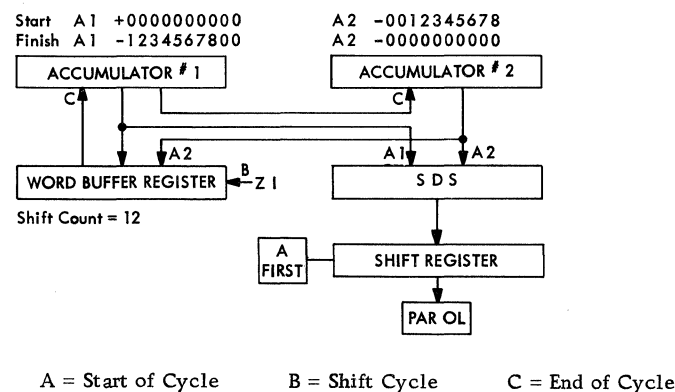


FIGURE 5.4-11. DOUBLE SHIFT LEFT AND COUNT--A1 ZERO

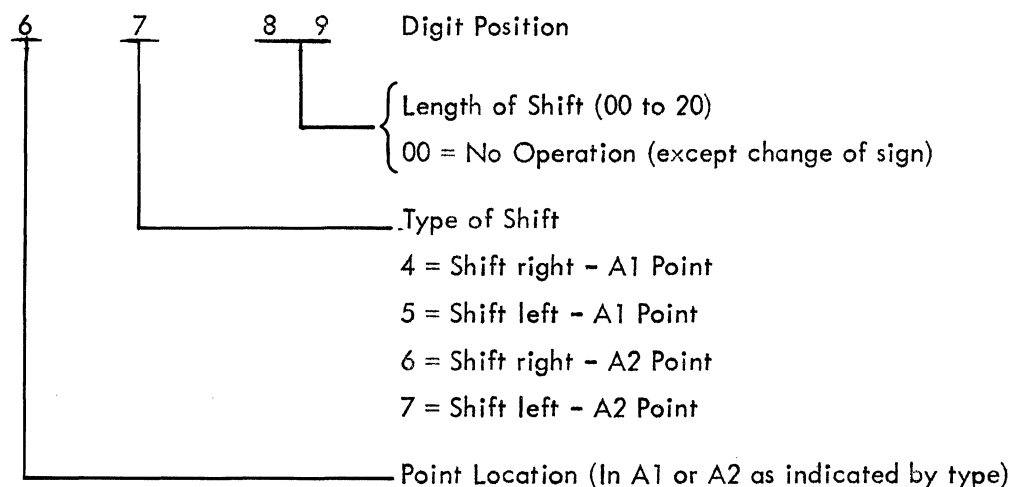
Shift Count Storage (Figure 5.4-4). The storage of the shift count is performed identical to the single-word operation. The program field register selects the index word, which enters the auxiliary register. The count value is entered as a four-digit value in positions 2 through 5 with an adder operation. The remainder of the word is unchanged. The modified word is stored back in its original address.

SPLIT SHIFTS

A second group of double-shift operations provide for shifting a portion of the factor leaving the remainder intact. The position of the split, the direction and amount of shift are specified in the address portion of the instruction word. The split is specified as a digit position rather than between two digits. In shifting, the point digit moves with the shift. This places the split to the right of the point digit on a left shift and to the left on a right shift. After the shift the space between the two portions is filled with zeros. Positions 6 - 9 of the instruction word may be indexed in the normal manner to adjust the instructions prior to the shift operation. The field control portion of the instruction word is not used for any of the split operations.

The signs are handled in the same manner as for a normal double shift, when the operation calls for a movement of digits in both accumulators. In the operations of right shift from accumulator 2 point and left shift from accumulator 1 point, only one accumulator is shifted and the signs are unchanged.

The address portion for a split shift is broken down as follows:



Each of the shift types breaks into two operation procedures dependent on whether the point digit moves out of its original factor word. This is determined as part of the setup cycle (Figure 5.4-12). Program register D is read out in parallel to the computer address bus to set its latches and the register is regenerated. For a right shift, the thousands position digit is set in the tens position of the program field register, while a nine is set in the units position. With left shift the units position is set from the latches, while the tens position is set to zero. These two settings, when used with the field ring, will count the positions to the right and left of the point digit, respectively. Determination of a shift out of the register limits is made by adding the point digit value from the program field register, and the shift value from program register D. The point digit value is complement added for left shift. An adder output of zero in the tens position indicates that the point digit will stay within the register, while a non-zero indicates a shift out. The unit's position adder output is used to set the shift register. The value indicates the amount of shift beyond the register on a shift-out operation. For other operations, the value is not used and is later reset.

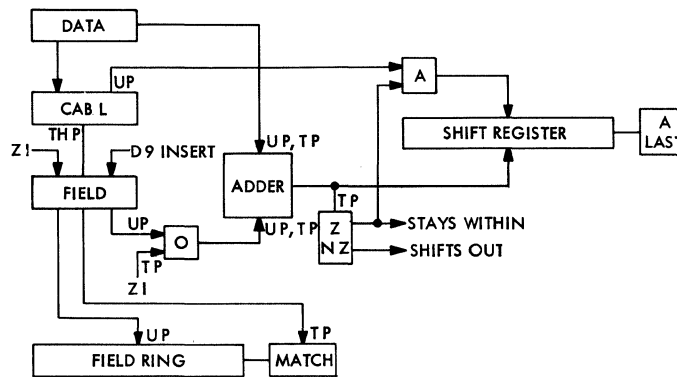


FIGURE 5.4-12. SPLIT SHIFT SET UP

Shift Right From Accumulator 1 Point

In a shift right from point in accumulator 1 operation, both accumulator 1 and 2 factors are used. The high-order positions above the point digit remain in the same positions at the end of the shift. The low-order position of accumulator 1 and all of accumulator 2 are right shifted by the shift value.

Point Digit Stays Within Accumulator 1 (Figure 5.4-13). This shift is the most difficult to make because accumulator 2 does not left shift, and three registers are needed to retain the required digits. On the first cycle, accumulator 1 and 2 are connected for right serial transfer with accumulator 1 as high order. Accumulator 2, position 9 is serially connected to position 9 of the word buffer register, which, with a right shift of accumulator 2, causes digits to enter the left shifting word buffer register in reverse order. Shifting is controlled by the field ring set and matched to the program field register. At the end of the cycle the point digit has just shifted out of accumulator 1 leaving those high order digits, which are to remain in their original positions at the end of shift. Before the end of the first cycle, the word buffer register is parallel-transferred to the auxiliary register, and accumulator 2 is parallel-transferred to the word buffer register.

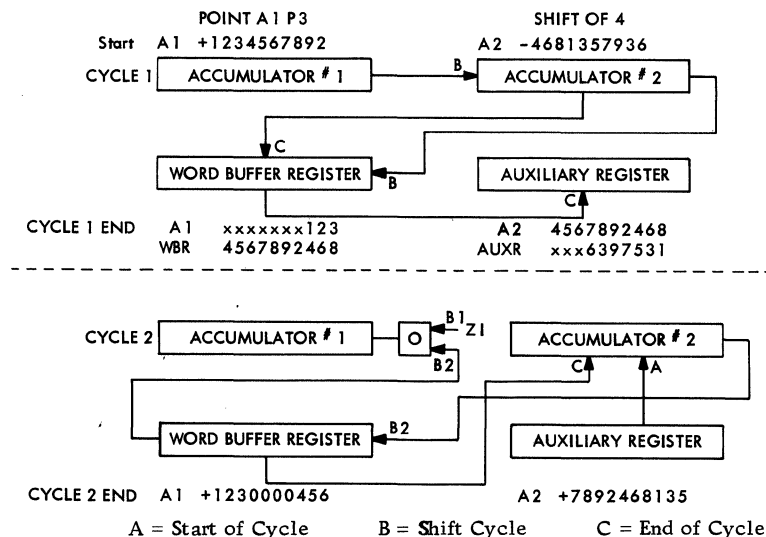


FIGURE 5.4-13. SHIFT RIGHT A1--POINT STAYS WITHIN A1

At the start of the second cycle, the reversed digits from the word buffer register are parallel-transferred from the auxiliary register to accumulator 2. The shift register is reset and set to the shift value from the computer address bus latches. For the first portion of the cycle, accumulator 1 is left shifted under control of the field ring. Zeros are inserted at position 9 under control of the shift register. The shift register is right shifted until A-last. At A-last, accumulator 1 and the word buffer register are connected for a left serial shift with accumulator 1 as high order. Again accumulator 2 is connected to right shift into the left shifting word buffer register. The digits, which were previously reversed, return to their original order and relative position. The shift continues until stopped by the field ring indicating that the accumulator 1 factor is back to its original position. The word buffer register with the remainder of the digits is parallel-transferred to accumulator 2 to end the operation.

Point Digit Shifts Out of Accumulator (Figure 5.4-14). When the point digit shifts out of accumulator 1, it is not necessary to transfer the digit back into that register on the second cycle. The remaining digits in accumulator 2 are further shifted to the right. The first cycle is performed in the same manner as for a point within.

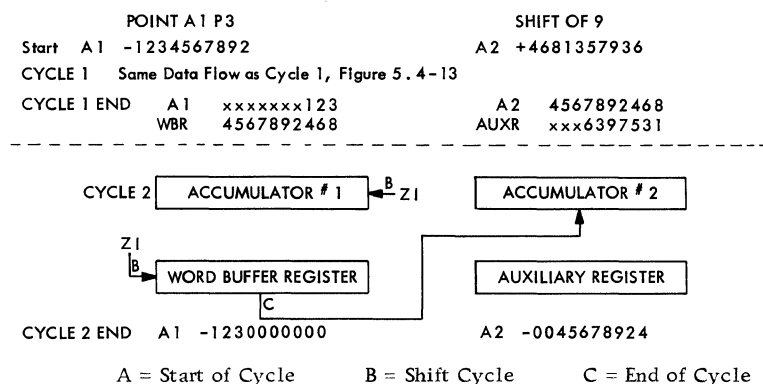


FIGURE 5.4-14. SHIFT RIGHT A1--POINT SHIFTS OUT OF A1

On the second cycle accumulator 1 is left shifted under control of the field ring. Zeros are inserted at position 9 for the full shift. The shift stops with a field ring match, when the accumulator 1 factor is back to its original position. The word buffer register containing the data from accumulator 2 is right shifted under control of the shift register. Zeros are inserted at position 0. The shift register retains the shift value set into it during the setup for this operation. It is right shifted to A-last to stop the word buffer register shift. At the end of the cycle, the word buffer register is parallel-transferred to accumulator 2 to end the operation.

Shift Left from Accumulator 1 Point

In a shift left from point in accumulator 1 operation, only the accumulator 1 factor is used. The low-order digits below the point remain in the same positions at the end of the shift. The remainder of the accumulator 1 digits are left shifted by the shift value.

Point Digit Stays Within Accumulator 1 (Figure 5.4-15). On the first cycle accumulator 1 is parallel-transferred to the word buffer register. Accumulator 1 and the word buffer register are connected for left serial shift with accumulator 1 as high order. The registers are left shifted under control of the field ring. At the end of the shift, the word buffer register contains only the digits to the right of the point digit.

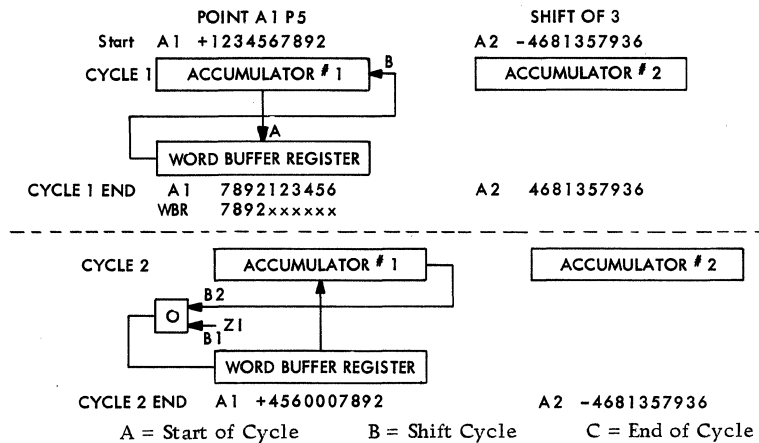


FIGURE 5.4-15. SHIFT LEFT A1--POINT STAYS WITHIN A1

At the start of the second cycle, the shift register is reset and set with the shift value from the computer address bus latches. The word buffer register right shifts under control of the field ring. Zeros are inserted into position 0 until the shift register reaches A-last. At A-last, the accumulator 1 and the word buffer register are connected for right serial shift with accumulator 1 as high order. The shift continues, returning digits from accumulator 1, until the field ring reaches the match point. At the end of the shift, the word buffer register with the final shifted factor is parallel-transferred to accumulator 1 to end the operation. Accumulator 2 is not changed.

Point Digit Shift Out of Accumulator (Figure 5.4-16). With the point left shifted out of accumulator 1, it is not necessary to replace any digits on the second cycle. All digits shifted out are lost. The first cycle is performed in the same manner as for a point within. On the second cycle, the word buffer register is right shifted under control of the field ring. Zeros are inserted at position 0 for the full shift. The shift stops when the field ring reaches the match position. At the completion of shift, the word buffer register is parallel-transferred to accumulator 1 to end the operation. Accumulator 2 is not changed.

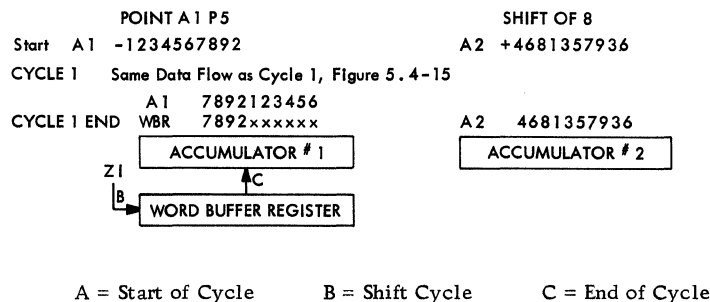


FIGURE 5.4-16. SHIFT LEFT A1--POINT SHIFTS OUT OF A1

Shift Right From Accumulator 2 Point

In a shift right from a point in accumulator 2 operation, only the accumulator 2 factor is used. The high-order digits above the point remain in the same positions at the end of the shift. The remainder of the accumulator 2 digits are right shifted by the shift value.

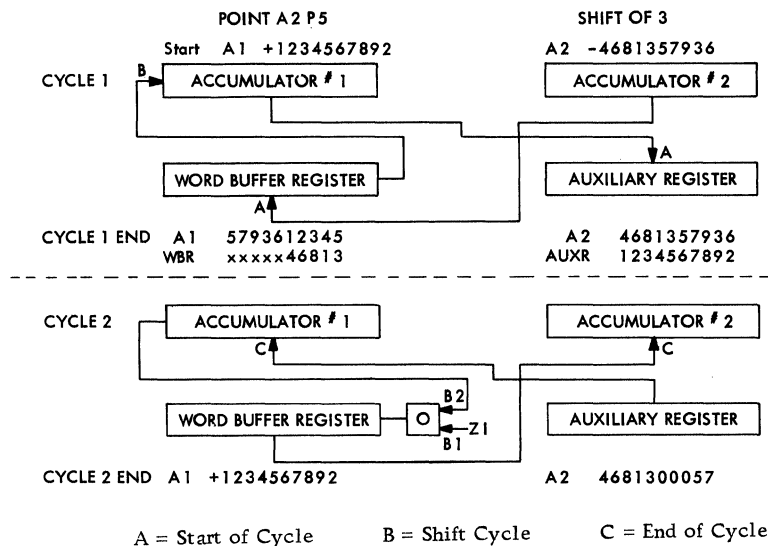


FIGURE 5.4-17. SHIFT RIGHT A2--POINT STAYS WITHIN A2

Point Digit Stays Within Accumulator 2 (Figure 5.4-17). At the start of the operation, the accumulator 1 factor is parallel-transferred to the auxiliary register to make use of the left shift feature of Accumulator 1. Accumulator 2 is parallel-transferred to the word buffer register for the operation. Accumulator 1 and the word buffer register are connected for right serial shift with the word buffer register as high order. The registers right shift under control of the field ring until match time. At the end of the shift, the word buffer register contains only the digits to the left of the point digit.

At the start of the second cycle, the shift register is reset and set with the shift value from the computer address bus latches. The word buffer register left shifts under control of the field ring. Zeros are inserted at position 9 until the shift register reaches A-last. At A-last, accumulator 1 and the word buffer register are connected for left serial shift with the word buffer register as high order. The shift continues, returning digits from accumulator 1, until the field ring reaches the match point. At the end of the shift, the word buffer register with the final shifted factor is parallel-transferred to accumulator 2. The original accumulator 1 factor is returned from the auxiliary register by parallel transfer, without change, to end the operation.

Point Digit Shifts Out of Accumulator 2 (Figure 5.4-18). With the point right shifted out of accumulator 2, it is not necessary to replace any digits on the second cycle. All digits shifted out are lost. The first cycle is performed in the same manner as for a point within. On the second cycle, the word buffer register is left shifted under con-

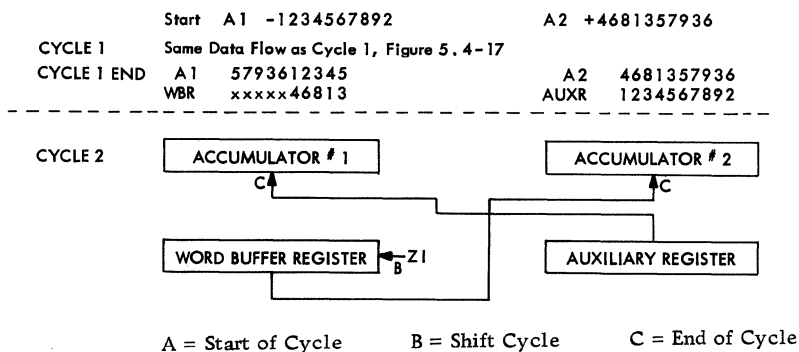


FIGURE 5.4-18. SHIFT RIGHT A2--POINT SHIFTS OUT OF A2

trol of the field ring. Zeros are inserted at position 9 for the full shift. The shift stops when the field ring reaches the match position. At the completion of shift, the word buffer register is parallel transferred to accumulator 2. The original accumulator 1 factor is returned from the auxiliary register by parallel-transfer, without change, to end the operation.

Shift Left From Accumulator 2 Point

In a shift left from a point in accumulator 2 operation, both accumulator 1 and 2 factors are used. The low-order positions below the point digit remain in the same positions at the end of the shift. The high order positions of accumulator 2 and all of of accumulator 1 are left shifted by the shift value.

Point Digit Stays Within Accumulator 2 (Figure 5.4-19). At the start of the first cycle, accumulator 2 is parallel-transferred to the word buffer register. Accumulator 1 and the word buffer register are connected for left serial shift at both ends. Digits shifted out of the position 0 end of each register are fed to the position 9 end of the other register like a serial regeneration operation. Shifting is under control of the field ring which continues until the match point. At this time, the point digit has just been shifted out of position 0 of the word buffer register. At the end of the first cycle, the word buffer register is parallel-transferred to accumulator 2 and regenerated.

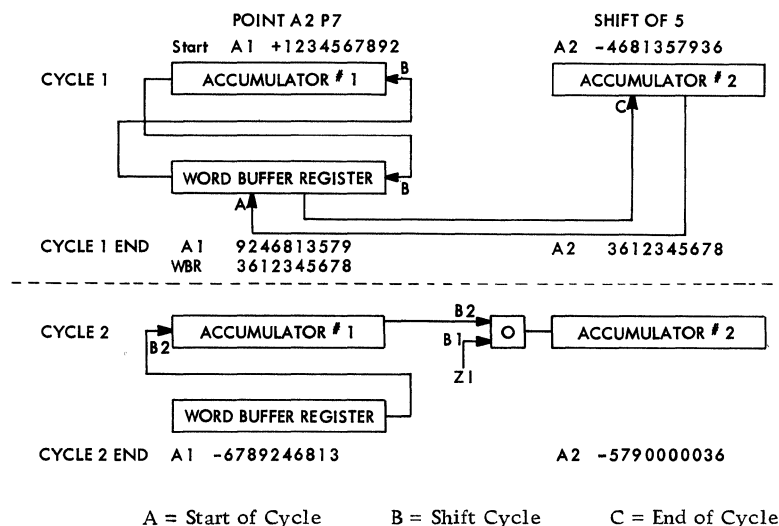


FIGURE 5.4-19. SHIFT LEFT A2--POINT STAYS WITHIN A2

At the start of the second cycle, accumulator 2 is right shifted under control of the field ring. Zeros are inserted at position 0 under control of the shift register which has been reset and set to the shift value. When the shift register reaches A-last, the zero insertion is stopped. The word buffer register, accumulator 1, and accumulator 2 are connected in that order for right serial transfer to move the shifted digits back within accumulator 2. The shift continues until the field ring match point to end the operation.

Point Digit Shifts Out of Accumulator 2 (Figure 5.4-20). With the point shifted out of accumulator 2, no digits need be shifted back into the register on the second cycle. The remaining digits in accumulator 1 are further shifted to the left. The first cycle is performed in the same manner as for a point within.

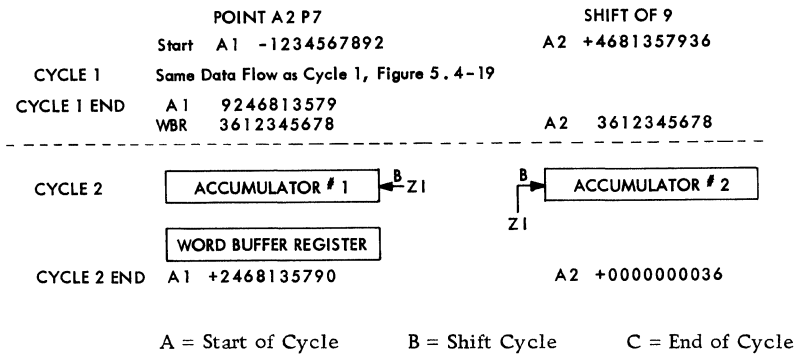


FIGURE 5.4-20. SHIFT LEFT A2--POINT SHIFTS OUT OF A2

On the second cycle accumulator 2 is right shifted with zeros inserted at position 0. The shift, under control of the field ring, continues until the match point, when the low-order portion of the factor is in its original position. Accumulator 1 is left shifted with zeros inserted at position 9. Accumulator 1 shifting is under control of the shift register, which retains the shift value set into it during the operation setup. It is right shifted to A-last to stop the operation. Both registers are shifted simultaneously with the operation ending following the longer shift.

5.4.01 +50 Shift Single

The initial setup for each of the single-shift operations is the same. The program data register is first read out in parallel to the computer address bus latches to provide a static read-out of the information. The latch output is used to determine the augmented operation code. In the case of shift single, where only one word is processed, the thousands position indicates the operated accumulator. The hundreds position indicates the type of shift. The tens and units positions denote the specified shift value except when a count is indicated.

In each of the shift-single operations, the operated accumulator is transferred to the word buffer register for the shift. The word buffer register is selected because of its ability to shift both left and right. The use of a separate register also permits a more simple switching for the round-and-count operations.

The type of shift signals developed from decoding the CAB latches are mixed to produce several common gating signals. The conditions selected allow one signal to perform a similar function in each of the shift types. Only the basic operation signals are listed below. The more specific mixes used for the individual shift types are detailed later under the operation headings.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Prg Ring Rcy Ctrl	C0	2F-41.12.03
RO PR Data For Sh	Prg Rcy	3B-4B-45.12.01
PR Data RO CAB	Prg Rcy	5G-42.02.04

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set CAB L Reg	CX AP	4H-5F-42.05.01
Force OP A 1	CX	2B-3B-42.11.26
Force OP A 2	CX	2B-3C-42.11.26
Force OP A 3	CX	2B-3D-42.11.26
Sh Code Sgl	CX	2B-42.11.24
Sh Code	CX	2B-42.11.32
Sh Rt Sgl	CX	4B-42.11.24
Sh Lt Sgl	CX	4F-42.11.24
Sh Rt Rnd Sgl	CX	4D-42.11.24
Sh Lt Cnt Sgl	CX	4H-42.11.24

Shift Right Single (XOXX) (Figure 5.4-21)

The shift right single operation is performed in a single program cycle. This is a continuation of the cycle in which the operation analysis was performed. At C0 time the operated accumulator is read out to the arithmetic bus and into the word buffer register for shifting. At the same time, the units position of the CAB latches is read into the cores of the shift register with a right shift. The shift register continues to right shift until the bit reads out to set the A-last latch. The word buffer register is simultaneously shifted to the right until stopped by the shift stop signal, which is set by A-last +. During the shift, zeros are inserted at position 0 of the word buffer register to fill the vacancies.

The program cycle ring is started at the last + position by switching A field end, set by the A-last signal, and a forced Athr Field End. The setting is inhibited until C2 to allow time for updating the instruction counter. The program ring is stopped at program cycle test in the normal manner. The shifted factor in the word buffer register is read out to the arithmetic bus at PC stop. It is read back into the operated accumulator to replace the original value. The operation is ended by developing a shift code end pulse at PC Last +.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rey	4B-5B-45.12.02
RT Sgl Sh Codes	CX	3A-42.11.27
Sgl 1 Cy Sh Codes	CX	3H-42.11.27

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
No Count On Sh	Sh Cy 1	5F-42.11.27
Sh RT Codes Cy 1	Sh Cy 1	3D-42.11.33
1 Cy Sh Codes	CX	3E-42.11.33
Shift Register Operation		
Sh Reg RI CAB Up	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A-Last +	4B-5B-45.12.03
Factor Shift		
A 1 RO AB	CX-0	2D-44.11.01
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Ser Sh RT	C0-1	2A-4B-5B-44.08.07
Zero Ins Psn 0	C1	4A-44.08.19
Reset WBR RT Sh	Sh Stop AP	3C-4C-44.08.07
WBR AB RO	PC Stop	2F-44.08.12
A 1 AB RI Ctrl	PC Stop	3C-44.11.00
End Operation		
IC Update Over	C2	4B-5B-45.12.07
A Field End	A-Last	3G-4E-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	A-Last CP	2F-4G-5G-42.40.22
Sh Code End	PC Last +	4E-45.12.03
End Data Op Mix	PC Last +	4F-42.40.30

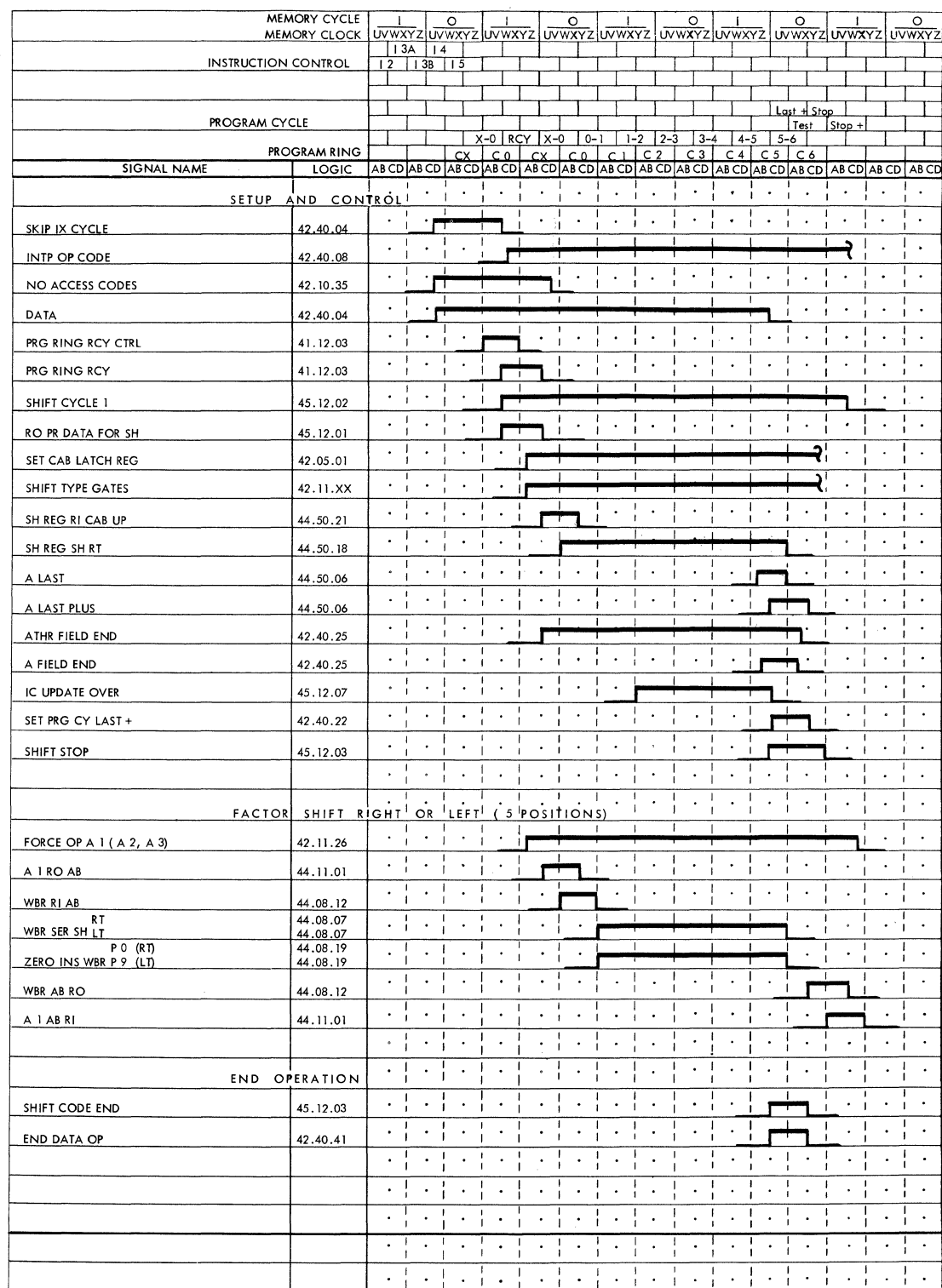


FIGURE 5.4-21. SHIFT 1 CYCLE SET UP--RIGHT SINGLE, LEFT SINGLE

Shift Left Single (X2XX) (Figure 5.4-21)

The shift left single operation is performed in a single program cycle in the same manner as a shift right operation. The operation differs only in the direction of the shift. Zeros are inserted at position 9 of the word buffer register instead of position 0 to fill the vacancies. The same timing pulses are used to start and stop the operation, but the operation code switching develops the gates for left shift. The shift register still shifts to the right for the shift count-down and sets the A-last latch. The program cycle ring and the shift code end controls are identical to the right shift operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sgl 1 Cy Sh Codes	CX	3G-42.11.27
No Cnt On Sh	Sh Cy 1	5G-42.11.27
Sh Lt Cy 1	Sh Cy 1	3A-42.11.33
1 Cy Sh Codes	CX	3E-42.11.33
Shift Register Operation		
Sh Reg RI CAB UP	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A-Last +	4B-5B-45.12.03
Factor Shift		
A 1 RO AB	CX-0	2D-44.11.01
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Ser Sh Lt	C0-1	3F-4F-5F-44.08.07
Zero Ins Psn 9	C1	4J-44.08.19
Reset WBR Lt Sh	Sh Stop AP	3G-4G-44.08.07
WBR AB RO	PC Stop	2F-44.08.12
A 1 AB RI Ctrl	PC Stop	3C-44.11.00

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Operation		
IC Update Over	C2	4B-5B-45.12.07
A Fld End	A-Last	3G-4E-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	A-Last CP	2F-4G-5G-42.40.22
Sh Code End	PC Last +	4E-45.12.03
End Data Op Mix	PC Last +	4F-42.40.30

Shift Right and Round Single (XIXX) (Figure 5.4-22)

The shift-right and round-single operation requires two program cycles to complete. The first cycle is identified with the shifting portion of the operation. The cycle is similar to the shift-only code except that the factor is not transferred back to the operated accumulator. The initial setup, the shift register control, the factor shift, and zero insert are controlled by the same gates as the simple right shift. At the end of the cycle, instead of developing the shift-end pulse, the program ring is recycled and the shift cycle 2 latch is set.

At the start of the second cycle the word buffer register contains the shifted factor, while the output latches were forced to retain the last digit that was shifted out. These eleven digits are serially fed to the adder on channel 1. The adder is also fed from the channel 2 insert with a five followed by ten zeros. The adder entries are gated with true-add and no-carry insert. The eleven-digit adder output is serially returned to the high order of the operated accumulator. The digits right shift across the register to replace the original value. The low order or adjusted digit is read out to its output latches on the last shift and is lost.

A call for a zero shift is treated as a simple shift right. The rounding cannot be done because the factor contains no digit to the right which can be adjusted.

On the first cycle, the program cycle ring is started at the Last + position by switching A-field end and a forced Athr Field End. The program ring is stopped and restarted by the recycle control. For the second cycle, the program cycle ring is started at Last + by forcing both A-field end and Athr Field End at the completion of the adding cycle. The program ring is stopped at program cycle test in the normal manner. The operation is ended by developing a shift-code-end pulse at PC Stop.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sh Cycle 2	Prg Rcy (2)	4F-5F-45.12.02
RT Sgl Sh Codes	CX	3B-42.11.27
Sgl 2 Cy Sh Codes	CX	3F-42.11.27
2 Cy Sh Codes	CX	3G-42.11.33
Sh RT Rnd	CX	3D-42.11.34
Sh Cy 2 RT Rnd	Sh Cy 2	4G-42.11.34
No Cnt On Sh	Sh Cy 1	5F-42.11.27
Shift Register Operation		
Sh Reg RI CAB UP	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A-Last +	4B-5B-45.12.03
A Fld End Set	CX	4G-42.40.33
A Fld End	A-Last	4E-5E-42.40.25
Factor Shift		
A 1 RO AB	CX-0	2D-44.11.01
WBR RI AB Ctrl	CX-0	2D-44.11.01
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Ser Sh RT	C0-1	2A-4B-5B-44.08.07
Zero Ins Pos 0	C1	4A-44.08.19
Reset WBR RT Sh	Sh Stop AP	3C-4C-44.08.07
Reset P 9 WBR Out L	PC Last +	4C-5C-6D-44.08.20

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Adder Cycle		
RO WBR Ch 1	CX	4F-5F-44.08.20
WBR Ser Sh RT	C0	2D-4B-5B-44.08.07
WBR Sh RT Turn Off	C10	3J-44.08.37
Ch 2 Ins 5 (3 Bit)	CX	4C-44.46.20
Ch 2 Ins Zero	C0	4F-5F-42.62.15
Ch 2 Ins Zero Reset	C9-10 (cy 2)	4H-4G-42.62.15
Turn On True Add	CX	4B-44.41.16
Turn Off True Add	PC Test	4G-44.41.16
True NC Insert	PC Stop (cy 1)	2C-44.41.03
RO Adder	CX CP	4A-5B-44.41.05
Accumulator Read-In		
A 1 Sh RT L Turn On	C0-1	6A-44.11.02
A 1 RI Adder On	C1	3B-3D-4D-44.11.07
A 1 Sh RT L Reset	PC Stop	6D-44.11.02
End Operation		
A Fld End	C9	4J-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	C9	2F-4G-5G-42.40.22
Sh Code End	PC Stop	4F-45.12.03
End Data Op Mix	PC Stop	4F-42.40.30

Shift Left and Count (X3XX) (Figures 5.4-23, 24)

The shift-left-and-count operation requires two program cycles along with time for data and store request cycles. The first cycle differs from the shift-left operation in that the amount of shift is not specified. During the transfer of the factor to the word buffer register, the significant digit scanner is turned on to sense the loca-

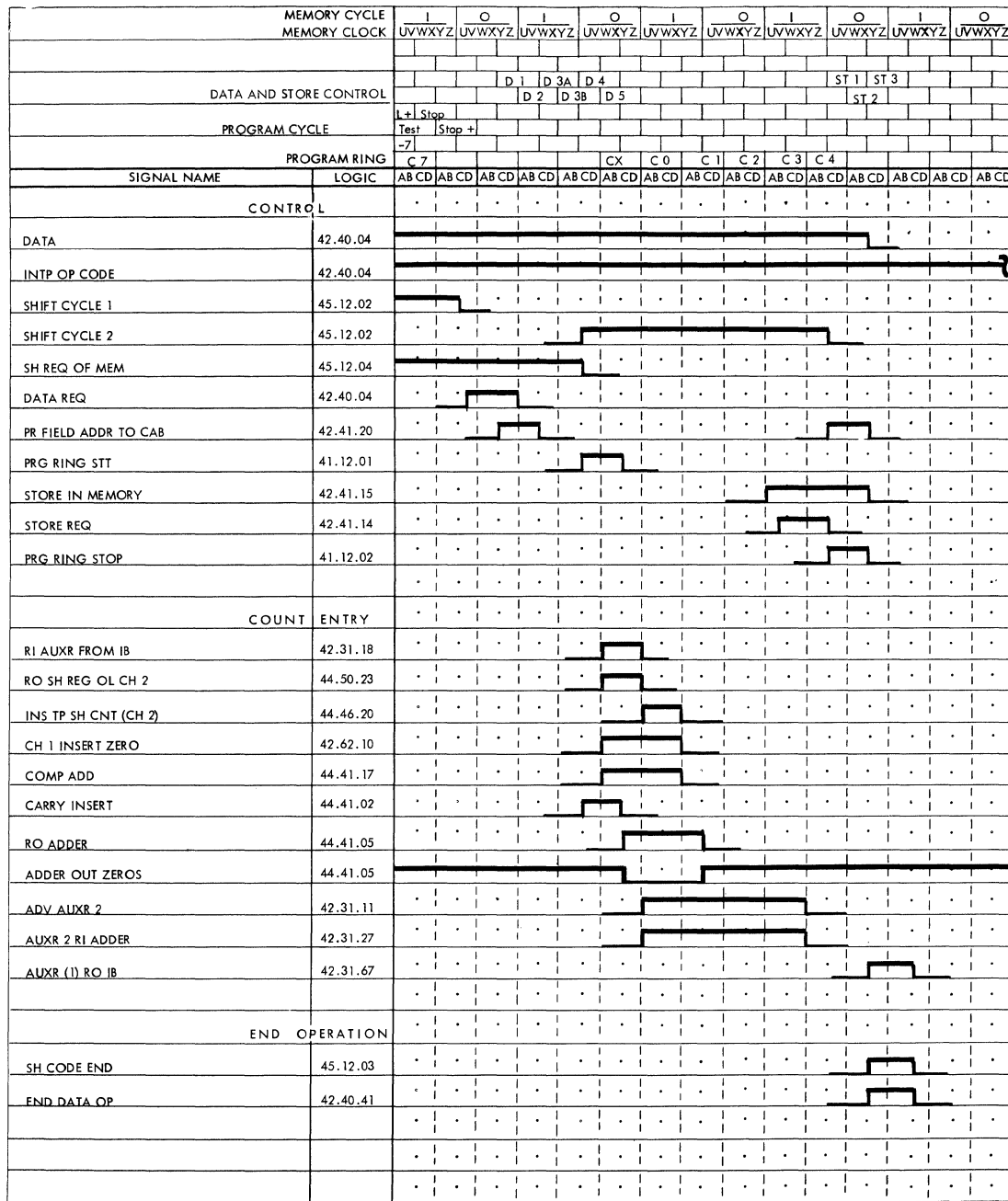


FIGURE 5.4-24. SHIFT LEFT COUNT--COUNT STORAGE CYCLE
SINGLE AND DOUBLE

with a left shift to start the count-down. The output latches remain set to allow recording the shift value. The register continues to left shift until the bit reads out of the high-order position to set the A-first latch and the A-field end latch. If no shift is indicated by the immediate setting of the A-first latch, a zero is forced into the output latches.

The left shift of the word buffer register is similar to the shift-left operation except that it is gated with shift left count signals. The shift starts three digits later due to the time required to set the shift register, and to read out to the A-first latch. The program cycle ring is started at Last + to return the shifted factor to the operated accumulator. The program ring is stopped in the normal manner at program cycle test.

At the end of the shift operation, the shift request of memory latch is set. In turn, data request and memory request are developed. The address used for this operation is that of the index word specified by the program field register. The index word is read into the auxiliary register from the information bus. The complement values from the shift register and the tens digit count latches are fed to channel 2. Zeros are fed to the channel 1 entry. The adder entries are gated with complement-add and carry insert to produce a true value output. The two adder output digits followed by two zeros are serially fed to the high order of Auxiliary 2 to replace the previous value. The remainder of the index word is not changed; however, the sign is set plus. The word is returned to core storage by setting the store in memory latch. In turn, store request and memory request are developed. The program field register is again used to place the word back in its original location.

The program ring is started at data control 4 on the second cycle. The shift cycle 2 latch is also set at this time. With the transfer of the address to core storage, the CAB latches are not reset, and thus the type of shift is retained. The program cycle ring is not started for this cycle. The program ring stop is forced at C4. The store request is initiated at C3. The operation is ended by developing a shift-code-end pulse at store control 3.

Variations exist in the count value controls, due to carry during complement conversion for storage. This problem occurs when the count value is either zero or ten. The zero is detected when the significant digit scanner reads out from the high-order position. In transferring to the shift register, the bit reads out the A-first latch immediately. With the sensing of this condition, a zero is forced into the shift register parallel output latches for the count value. This condition also forces the setting of the zero insert and the reset of the normal nine insert. The output latch zero enters the adder as a nine, which adds to the carry insert to produce a zero output and a carry. The tens zero insert enters as a nine, adds to the carry, and reads out a zero. The carry from the tens position is ignored.

A count value of ten occurs when the significant digit scanner sets its zero factor latch. With this condition the shift register parallel output latches are set to zero, and the A-field end latch is forced to turn on the program cycle. The shift register zero reads into the adder as a nine, adds to the carry insert, and reads out a zero and a carry. The tens digit nine insert reads into the adder as a zero, adds a zero, adds to the carry and becomes a one.

<u>Major Signals</u>	<u>Key Signals</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sh Cycle 2	DC4	2F-5F-45.12.02
Sgl 2 Cy Sh Codes	CX	3E-42.11.27
Sh Code Cy 1	Sh Cy 1	3C-42.11.32
Sh Lt Cnt	CX	3H-42.11.32

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh Lt Cnt Cy 1	Sh Cy 1	5H-42. 11. 32
Sh Lt Cnt Cy 2	Sh Cy 2	5G-42. 11. 32
2 Cy Sh Codes	CX	3G-42. 11. 33
Factor Shift		
A 1 RO AB	CX-0	2D-44. 11. 01
SDS On	CX-0	2A-44. 06. 09
WBR RI AB Ctrl	CX-0	3A-44. 08. 12
WBR Ser Sh Lt	C2-3	3H-4F-5F-44. 08. 07
Zero Ins Pos 9 (WBR)	C3	44. 08. 19
Reset WBR RT Sh	Sh Stop AP	4H-4G-44. 08. 07
WBR AB RO	PC Stop	2F-44. 08. 12
A 1 AB RI Ctrl	PC Stop	3C-44. 11. 00
Shift Register Operation		
Sh Reg Par Regen		3B-44. 50. 19
SDS Reg Zero	C1 AP	3B-44. 06. 05
SDS Zero On Sh Cnt	C1	4B-5B-45. 12. 08
Sh Reg Par RO Ctrl	C1-2	5C-44. 50. 19
ZI On Sh (0 Sh)	C1	5G-44. 50. 23
ZI On Sh (10 Sh)	C2	5H-44. 50. 23
Block Ser RO Ctrl (RI)	C2-3	2G-44. 50. 19
Sh Reg Sh Lt Ctrl	C2-3	3F-4F-5F-44. 50. 18
A-First	AP	3A-4B-5B-44. 50. 22
A Fld End Set	CX	4G-42. 40. 33
A Fld End (Normal)	A-First	4E-5E-42. 40. 25
A Fld End (SDS Zero)	C1	3A-5E-42. 40. 25

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Prg Cy Last +	A Fld End	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-45. 12. 03
Ins Nine On Sh Cnt	CX	4B-5B-45. 12. 05
Ins Zero On Sh Cnt	C1	2G-5H-45. 12. 05
Read-In Index Word		
Sh Req of Mem	PC Last +	4B-5B-45. 12. 04
Data Req	MOP	3H-4H-5G-42. 40. 04
Prevent CAB L Clear	PC Last +	4G-5G-42. 11. 24
Start Data Ctrl Ring	Addr Gate	3B-4B-5B-42. 41. 10
PR Fld Addr to CAB	DC1	4H-42. 41. 20
Prg Ring Stt	DC4	4D-41. 12. 01
RI Auxr From IB	DC4	3D-5G-6G-42. 31. 18
Adder Operation		
RO Sh Reg OL Ch 2	CX	5F-44. 50. 23
Ins TP Sh Cnt (9)	C0	5B-44. 46. 20
Ch 1 Insert 0	DC4	3G-4B-42. 62. 10
Turn On Comp Add	DC4	4D-44. 41. 14
Reset Comp Add	C1	4G-5G-44. 41. 17
Carry Insert	CX AP	3H-44. 41. 02
RO Adder	CX CP	3C-4B-5B-44. 41. 05
Adv Auxr 2	C0	4B-5E-42. 31. 11
Adv Stop (Auxr 2)	C4	5C-42. 31. 50
Auxr 2 RI Adder	C0	4B-5A-42. 31. 27
Stp Auxr 2 RI Adder	C4	5B-6A-42. 31. 62
End Operation		
Store in Memory	C3	3F-4G-42. 41. 15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Store Req	MIP	2B-3B-42.41.14
Prg Ring Stop	C4 CP	3G-4G-5G-41.12.02
Sh Code End	STC3	4G-45.12.03
End Data Op Mix	STC3	4F-42.40.30

5.4.02 - 50 Shift Double - Full Factor

The normal shift-double codes are similar to their shift-single counterparts. Instead of specifying an accumulator for the operation, the shift-double codes always use accumulators 1 and 2 in tandem. The initial setup to start the operation and analysis of the augmented code is identical. The thousands position of the computer address bus latches must be zero for the normal codes. A digit in this position indicates the point for a split shift to be covered later. The hundreds position indicates the type of shift. The units and tens positions denote the amount of shift required except when a count is indicated. The shift may be as great as twenty positions. To save time, each of the operations has control variations to process shifts of less than ten and greater than ten. On a shift of ten or greater, one complete register is lost, and the other register is filled with zeros.

In the normal shift-double codes the word buffer register and accumulator 1 are used together for the shift function. The two registers can be coupled for either register to serve as high order. The shift can be in either direction with provisions for zero insert. In several of the shift types, accumulator 2 is transferred to the word buffer register, which serves as the low order. For the basic count operation accumulator 1 is transferred to the word buffer register for the high order and accumulator 2 is transferred to accumulator 1.

The type of shift signals developed from decoding the hundreds position of the computer address bus latches are mixed to produce common gating signals. The conditions are selected to allow one signal to perform a similar function in each of the shift types. The control for a shift of ten, or greater, is developed from the tens position latches. Only the basic operation signals are listed as follows. The specific mixes used for the individual shift type are detailed under the type heading.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Prg Ring Recy Ctrl	C0	2F-41.12.03
RO PR Data for Sh	Prg Recy	3B-4B-45.12.01
PR Data RO CAB	Prg Recy	5G-42.02.04
Set CAB L Reg	CX AP	4H-5F-42.05.01
Dbl Sh Code	CX	2A-42.11.25

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh Code Dbl	CX	Buf-42. 11. 29
Sh Code	CX	2A-42. 11. 32
Sh RT Dbl	CX	4A-42. 11. 25
Sh Lt Dbl	CX	4C-42. 11. 25
Sh RT Rnd Dbl	CX	4B-42. 11. 25
Sh Lt Cnt Dbl	CX	4D-42. 11. 25
CAB L TP Dec 0	CX	3C-42. 11. 28
CAB L TP Dec 1	CX	2D-42. 11. 28
Sh 10 to 21	CX	2J-42. 11. 28
Zero Sh	CX	3B-3C-4B-42. 11. 28

Shift Right Double (00XX) (Figure 5.4-25)

The shift-right-double operation is performed in a single program cycle. This is a continuation of the cycle in which the operation analysis was performed. The word buffer register functions as the low-order register to handle the accumulator 2 portion of the factor. At the start of the operation, if the specified shift is less than ten, accumulator 2 is transferred to the word buffer register. If the shift is ten, or greater, accumulator 1 is immediately transferred to the word buffer register to effect a shift of ten. The accumulator to be transferred is selected by controlling the operated accumulator latches. Signals developed from the tens digit CAB latches are used for this purpose.

The selection of the register to be shifted is controlled by the size of the shift. With a shift of less than ten, both registers must be shifted, because a portion of the accumulator 2 factor (WBR) is shifted out and must be replaced by digits from accumulator 1. Zeros are inserted at the high order of accumulator 1. Shifts of ten, or greater, require shifting only the word buffer register, containing the accumulator 1 portion of the factor. Zeros are inserted in this case at the high order of the word buffer register. Accumulator 1 with a shift of ten, or greater, requires the insertion of ten zeros. These are obtained by forcing the arithmetic register to read in from its parallel zero insert latches and then transferring at the end of the shift. With either operation the the operated accumulator 2 latch is forced to read in the transfer from the word buffer register.

The control of the amount of shift is by the shift register as with shift-single operations. The shift value enters the shift register from the units position of the CAB latches and then right shifts until the bit reads out to set the A-last latch and the A-field end latch. The factor registers are right shifted until stopped by the shift-stop signal at A-last +.

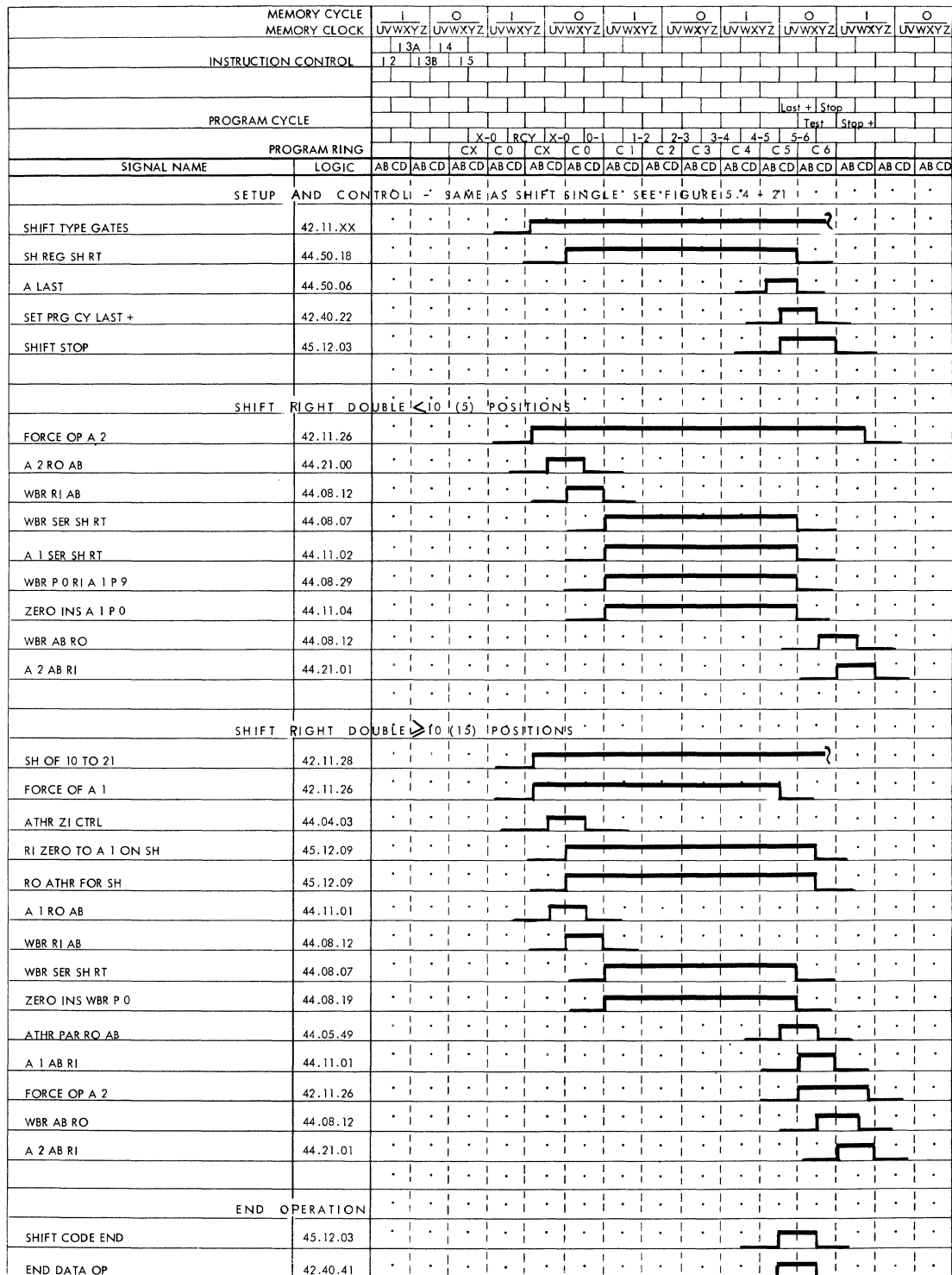


FIGURE 5.4-25. SHIFT RIGHT DOUBLE

The program cycle ring is started at the Last + position by switching A-field end and a forced Athr Field End. The setting is inhibited until C2 to allow time for updating the instruction counter. The program ring is stopped at program cycle test in the normal manner. The operation is ended by developing a shift code end pulse at program cycle last +.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
RT Dbl Sh Codes	CX	3C-42.11.29
Dbl 1 Cy Sh Codes	CX	3F-42.11.29
No Cnt on Sh	Sh Cy 1	5D-42.11.29
Sh RT Codes Cy 1	Sh Cy 1	3C-42.11.33
1 Cy Sh Codes	CX	3F-42.11.33
Shift Register Operation		
Sh Reg RI CAB UP	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A-Last +	4B-5B-45.12.03
Factor Shift Less Than Ten		
Force Op A 2	CX	3A-6G-42.11.26
A 2 RO AB	CX-0	3A-44.21.00
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh RT Turn On	C0-1	2A-44.08.07
A 1 Sh RT L Turn On	C0-1	2B-44.11.02
WBR P 0 RI A1P9	C1	4B-5B-44.08.29
Z Ins A 1 P 0	C1	3D-44.11.04
WBR AB RO	PC Stop	2F-44.08.12
A 2 AB RI Ctrl	PC Stop	2H-44.21.00
Factor Shift Ten or Greater		
Force Op A 1	CX	4E-5E-42.11.26
A 1 RO AB	CX-0	2D-44.11.01

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh RT Turn On	C0-1	2A-44.08.07
Z Ins Psn 0 (WBR)	C1	3B-44.08.19
Force Op A 2	PC Test	4G-6G-42.11.26
WBR AB RO	PC Stop	2F-44.08.12
A 2 AB RI Ctrl	PC Stop	2H-44.21.00
Athr Z I Ctrl	CX-0	3G-44.04.03
RI Zero to A 1 On Sh	C0	4B-5B-45.12.09
RO Athr For Sh	C0	5D-45.12.09
Athr Par RO to AB	PC Last +	4G-44.05.49
A 1 AB RI Ctrl	PC Last +	3H-44.11.00
End Operation		
IC Update Over	C2	4B-5B-45.12.07
A Fld End	A-Last	3G-4E-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	A-Last CP	2F-4G-5G-42.40.22
Adj Sign on RT Sh	Prg Stp	4B-5B-45.12.15
Sh Code End	PC Last +	4E-45.12.03
End Data Op Mix	PC Last +	4F-42.40.30

Shift Left Double (02XX) (Figure 5.4-26)

The shift-left double operation is performed in a single program cycle in a manner similar to the shift-right operation. The primary difference in the operation is in the direction of shift. The word buffer register functions as the low order register to handle the accumulator 2 portion of the factor. At the start of the operation, accumulator 2 always transfers to the word buffer register regardless of the shift value.

With a specified shift of less than ten, both registers, accumulator 1 and the word buffer register are shifted for the value of the units position of the shift value. Zeros

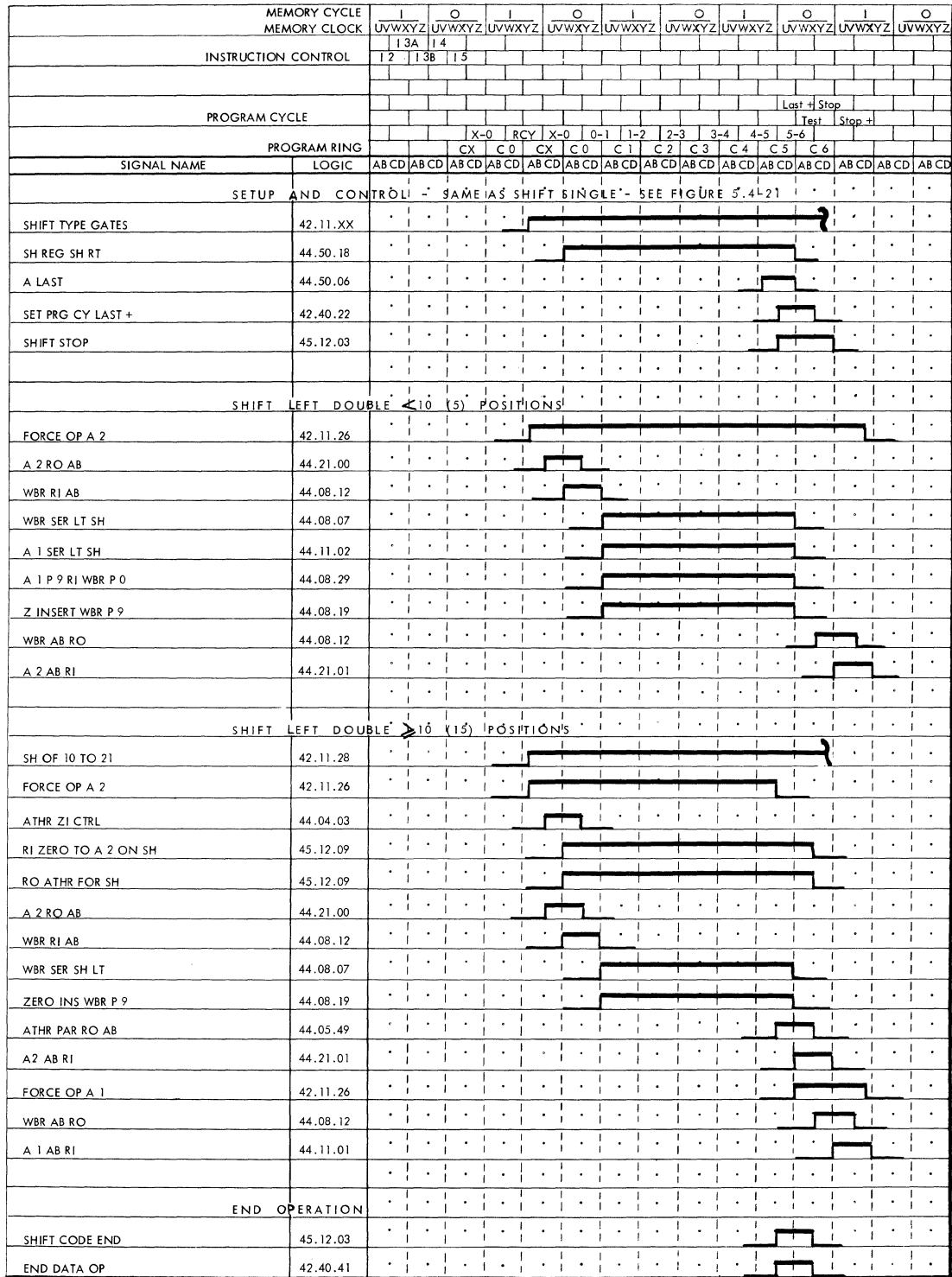


FIGURE 5.4-26. SHIFT LEFT DOUBLE

are entered at the low order of the word buffer register. At the end of the shift, the tens digit of the shift value selects operated accumulator 2 latch for the transfer. When the specified shift is ten, or greater, only the word buffer register is shifted, because the accumulator 1 value is shifted out. Zeros are entered at the low order of the register. The operated accumulator 1 latch is selected which causes A1 to read in the factor from the word buffer register to effect a shift of ten on the transfer. Accumulator 2 requires the insertion of ten zeros to fill the vacancies. These are obtained from the arithmetic register by forcing the read-in of the parallel zero insert latches.

The same basic timing pulses are used to control the operations as for a right shift. The left-shift operation code switching develops the gates for the difference in shifting and transfer. The shift register still shifts to the right for the count-down and sets the A-last latch. The program cycle ring and the shift code end controls are identical to the right-shift operation.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Lt Dbl Sh Codes	CX	3H-42.11.29
Dbl 1 Cy Sh Codes	CX	3E-42.11.29
No Cnt On Sh	Sh Cy 1	5E-42.11.29
Sh Lt Cy 1	Sh Cy 1	3B-42.11.33
1 Cy Sh Codes	CX	3F-42.11.33
Shift Register Operation		
Sh Reg RI CAB UP	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A Last +	4B-5B-45.12.03
Factor Shift Less Than Ten		
Force Op A 2	CX	2A-42.11.26
A 2 RO AB	CX-0	3A-44.21.00
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh Lt Turn On	C0-1	3F-44.08.07

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 1 Sh Lt Turn On	C0-1	5E-44.11.02
A 1 P 9 RI WBR P 0	C1	4F-44.11.08
Z Ins WBR P 9	C1	4J-44.08.19
WBR AB RO	PC Stop	2F-44.08.12
A 2 AB RI Ctrl	PC Stop	2H-44.21.00
Factor Shift Ten or Greater		
Force Op A 2	CX	5G-6G-42.11.26
A 2 RO AB	CX-0	3A-44.21.00
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh Lt Turn On	C0-1	3F-44.08.07
ZI WBR P 9	C1	4J-44.08.19
Force Op A 1	PC Test	3E-5E-42.11.26
WBR AB RO	PC Stop	2F-44.08.12
A 1 AB RI Ctrl	PC Stop	3C-44.11.00
Athr ZI Ctrl	CX-0	3G-44.04.03
RI Zero to A 2 On Sh	C0	4F-5F-45.12.09
RO Athr For Sh	C0	5E-45.12.09
Athr Par RO to AB	PC Last +	4G-44.05.49
A 2 AB RI Ctrl	PC Last +	5H-44.21.00
End Operation		
IC Update Over	C2	4B-5B-45.12.07
A Fld End	A Last	3G-4E-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	A Last CP	2F-4G-5G-42.40.22
Adj Sign on Lt Sh	Prg Stop	4G-5G-45.12.15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh Code End	PC Last +	4E-45.12.03
End Data Op Mix	PC Last +	4F-42.40.30

Shift Right and Round Double (01XX) (Figures 5.4-27A and 27B)

The shift-right and round-double operation requires three program cycles to complete. The first cycle is identified with the shift portion of the operation. This cycle is similar to the shift-only code except that the portion of the factor in the word buffer register is not transferred to accumulator 2. The setup, the shift register control, the factor shift, and the zero inserts are controlled by the same gates as for a simple right shift. At the end of the first cycle instead of developing the shift-end pulse, the program ring is recycled, and the Shift Cycle 2 latch is set.

The second and third program cycles are used in succession to pass the twenty-one digit factor through the adder to effect the rounding function. At the start of the second cycle, the word buffer register contains the low-order ten digits of the shifted factor. The register output latches were forced to retain the last digit that was shifted out. These eleven digits are serially fed to the adder on channel 1. During this cycle channel 2 is fed from the digit insert with a five followed by ten zeros. The adder entries are gated with true-add and no-carry insert. The eleven digit adder output is serially returned to the high order of accumulator 2 by forcing the operated accumulator 2 latch. The digits right shift across the register to replace the original value. The adjusted digit is read out to the output latches on the last shift and is lost. As the eleventh digit enters the adder, the program ring is recycled, and the shift cycle 3 latch is set.

During the third cycle accumulator 1 is right shifted to serially feed channel 2. The first digit immediately follows the last digit of the previous cycle. Zeros now feed on channel 2 from its insert, and true-add remains on.

It should be noted that channels 1 and 2 are alternated with the start of accumulator 1 readout. This allows use of common channel switching, which gates the word buffer register to channel 1 and accumulator 1 to channel 2. The carry condition from the final cycle 2 digit is used to gate the first entry. The ten-digit adder output returns to the high order of accumulator 1.

A call for a zero shift is treated as a simple shift-right-double operation. The rounding operation is inhibited because the factor contains no digit to the right for adjustment. For a shift of ten, the operation is forced to obtain a correction digit in the minimum time. Accumulator 2 is transferred to the word buffer register, where it is left shifted one digit to read out the high order to the output latch. The shift-stop circuits are forced to end the cycle. Accumulator 1 is transferred to the word buffer register for the high order ten digits. Zeros read into accumulator 1 on the start of the second cycle. Shift cycles 2 and 3 are performed in the normal manner.

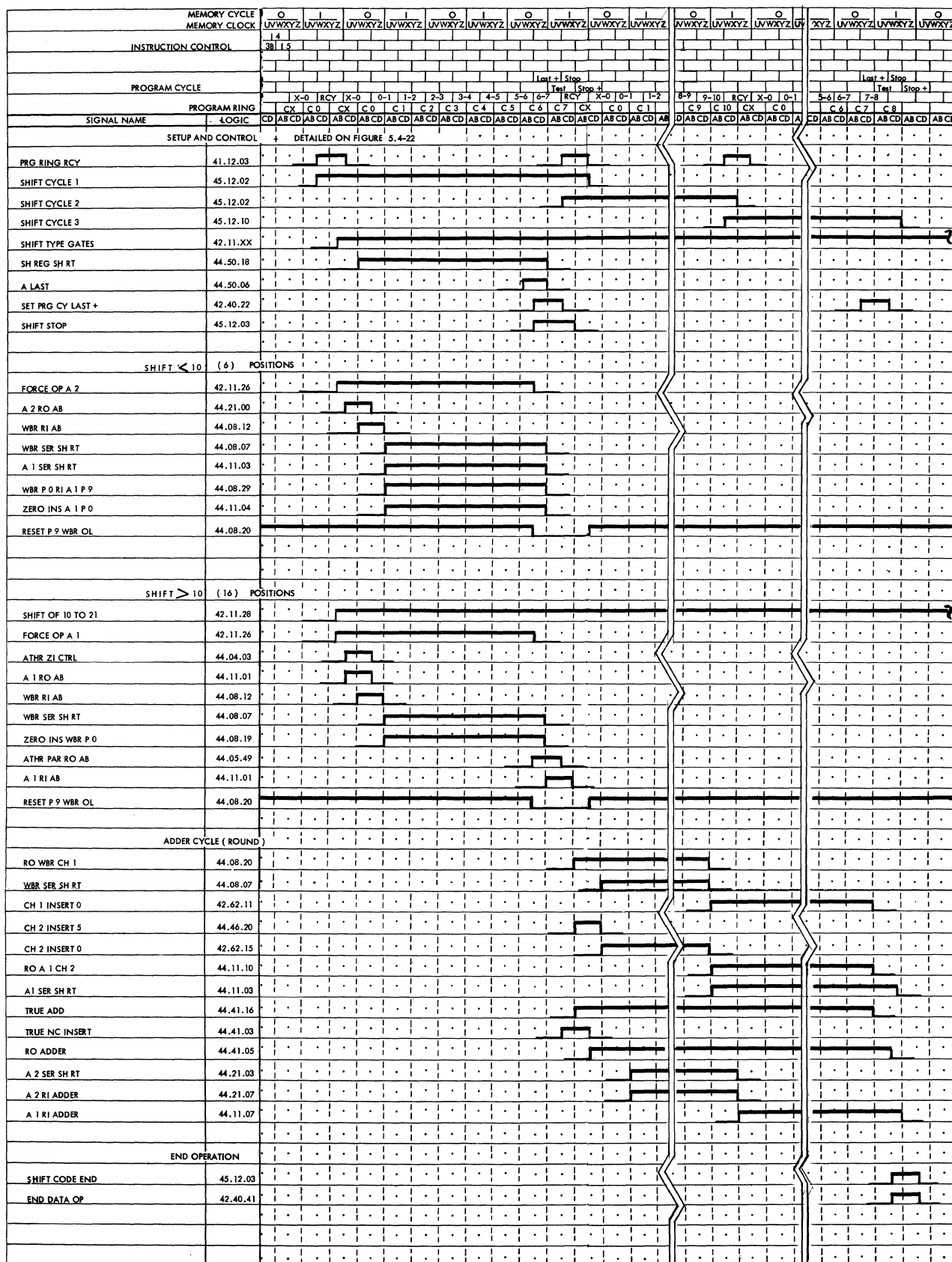


FIGURE 5.4-27A. SHIFT RIGHT ROUND DOUBLE

[illegible]

FIGURE 5.4-27B. SHIFT RIGHT ROUND DOUBLE--10 POSITIONS

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sh Cycle 2	Prg Rcy (2)	4F-5F-45.12.02
Sh Cycle 3	Prg Rcy (3)	4D-5C-45.12.10
RT Dbl Sh Codes	CX	3D-42.11.29
Dbl 2 Cy Sh Codes	CX	3J-44.11.29
No Cnt On Sh	Sh Cy 1	5D-44.11.29
Sh RT Codes Cy 1	Sh Cy 1	3C-44.11.33
2 Cy Sh Codes	CX	3H-44.11.33
Sh RT Rnd	CX	3E-44.11.34
Sh Cy 2 RT Rnd	Sh Cy 2	4G-44.11.34
Shift Register Operation		
Sh Reg RI CAB UP	CX-0	5B-44.50.21
Sh Reg Sh RT Ctrl	C0	2A-4C-5B-44.50.18
A-Last	AP	4B-5B-44.50.06
Sh Reg Sh RT Reset	A-Last AP	3D-4D-44.50.18
Sh Stop	A Last +	4B-5B-45.12.03
A Fld End Set	CX	4G-42.40.33
A Fld End		
Factor Shift, Less than Ten		
Force Op A 2	CX	3A-6G-42.11.26
A 2 RO AB	CX-0	3A-44.21.00
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh RT Turn On	C0-1	2A-44.08.07
A 1 Sh RT L Turn On	C0-1	2B-44.11.02
WBR P 0 RI A 1 Psn 9	C1	4B-5B-44.08.29

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Z Ins A 1 0 P	C1	3D-44.11.04
Reset P 9 WBR Out L	PC Last +	4C-5C-44.08.20
Factor Shift - Ten Only		
Sh RT Rnd Dbl 10	CX	4B-42.11.29
Reset Force Op A 1 L	CX	3H-4J-42.11.26
A 2 AB RO Ctrl	CX-0	5C-44.21.00
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Ser Sh Lt	C1	4H-44.08.07
A-Last	C0	2H-44.50.06
Reset P 9 WBR Out L	PC Last +	4C-5C-44.08.20
A 1 RO AB	PC Stop	4J-44.11.01
WBR RI AB	PC Stop	4D-44.08.12
Athr ZI Ctrl	CX-0 (cy 2)	3G-44.04.03
Athr Par RO to AB	C1-2 (cy 2)	5D-44.05.54
A 1 AB RI Ctrl	C1-2 (cy 2)	3E-44.11.00
Factor Shift, Greater than Ten		
Force OP A 1	CX	4E-5E-42.11.26
A 1 RO AB	CX-0	2D-44.11.01
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Sh RT Turn On	C0-1	2A-44.08.07
Z Ins Psn 0 (WBR)	C1	3B-44.08.19
Force Op A 2	PC Test	4G-6G-42.11.26
Reset P 9 WBR Out L	PC Last +	4C-5C-44.08.20
Athr ZI Ctrl	CX-0	3G-44.04.03
RI Zero to A 1 On Sh	C0	4B-5B-45.12.09
RO Athr for Sh	C0	5D-45.12.09

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Par RO to AB	PC Last +	4G-44.05.49
A 1 AB RI Ctrl	PC Last +	3H-44.11.00
Adder, Cycle 2		
RO WBR Ch 1	CX	4f-5F-44.08.20
WBR Ser Sh RT	C0	2D-4B-5B-44.08.07
WBR Sh RT Turn Off	C10	3J-44.08.37
Ch 2 Ins 5 (3 Bit)	CX	44.46.20
Ch 2 Ins Zero	C0	4F-5F-42.62.15
Ch 2 Ins Zero Reset	C9-10 (cy 2)	4H-4G-42.62.15
Turn On True Add	CX	4B-44.41.16
True NC Insert	PC Stop (cy 1)	2C-44.41.03
RO Adder	CX CP	4A-5B-44.41.05
A 2 Sh RT L Turn On	C0-1	6A-44.21.02
A 2 RI Adder	C1	3C-3D-44.21.07
A 2 Sh RT L Reset	CX (cy 3)	5E-44.21.02
Adder, Cycle 3		
RO A 1 Ch 2	C10 (cy 2)	4C-5F-44.11.10
A 1 Sh RT L Turn On	C9-10 (cy 2)	3C-44.11.02
Set Ch 1 Zero Ins	C10 (cy 2)	42.62.11
Reset Ch 1 Zero Ins	PC Test	42.62.11
Turn Off True Add	PC Test	4G-44.41.16
A 1 RI Adder On	Sh Cy 3 AP	4A-3D-44.11.07
A 1 Sh RT L Reset	PC Stop	6D-44.11.12

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
End Operation		
A Fld End	C7	3B-5E-42.40.25
Athr Fld End	CX-0	4A-5B-42.40.25
Set Prg Cy Last +	C7	2F-4G-5G-42.40.22
Adj Sign On RT Sh	Prg Stp	4B-5B-45.12.15
Sh Code End	PC Stop	4H-45.12.03
End Data Op Mix	PC Stop	4F-42.40.30

Shift Left and Count Double (03XX) (Figures 5.4-28A, 28B, and 28C)

The shift-left and count operation requires two program cycles along with data and store request cycles. The operation resembles the shift-left and count-single operation more than the shift-left-double. The shift value is determined by analysis of high order zeros to set the shift register instead of a specified value. The second cycle is used to store the count value in a specified index word.

At the start of the operation, accumulator 1 is transferred to the word buffer register. During the transfer the significant digit scanner is turned on to sense for the high-order significant digit. If a digit is sensed, the bit is transferred to the shift register for the shift value. Accumulator 2 is then read into accumulator 1 to serve as the low order for the shift. The two registers are shifted together and are coupled to read the high order of accumulator 1 into the low order of the word buffer register. Zeros are inserted at the low order of accumulator 1 to fill the vacancies.

If the significant digit scanner senses a zero factor in accumulator 1, the program ring is recycled, and the operation is restarted. The Double Count 10 or Up latch is now set causing the operated accumulator 1 latch to be reset and the latch for accumulator 2 to be set. Accumulator 2 is now transferred to the word buffer register to replace the previous zeros. The significant digit scanner is again turned on to check for a high-order digit. If a digit is sensed, the bit is transferred to the shift register for the shift value. Only the word buffer register is shifted leaving the zeros in accumulator 1. Zeros are inserted at the low order of the word buffer register to fill the vacancies.

If a zero factor is sensed during both register transfers, the shift register is forced to read in a zero shift value. The tens digit insert latches for the shift count are changed with each sensing by the significant digit scanner.

At the end of the shift operation, the factor in accumulator 1 is transferred to accumulator 2 and the word buffer register is transferred to accumulator 1. If accumulator 1 originally contained a significant factor, the effective shift is equal to the number of high order zeros with a count of less than ten. When accumulator 1 is all zeros, the shift is effectively ten positions plus the count of the zeros originally in accumulator 2. If both accumulators are zeros, the shifts and transfers are inhibited.

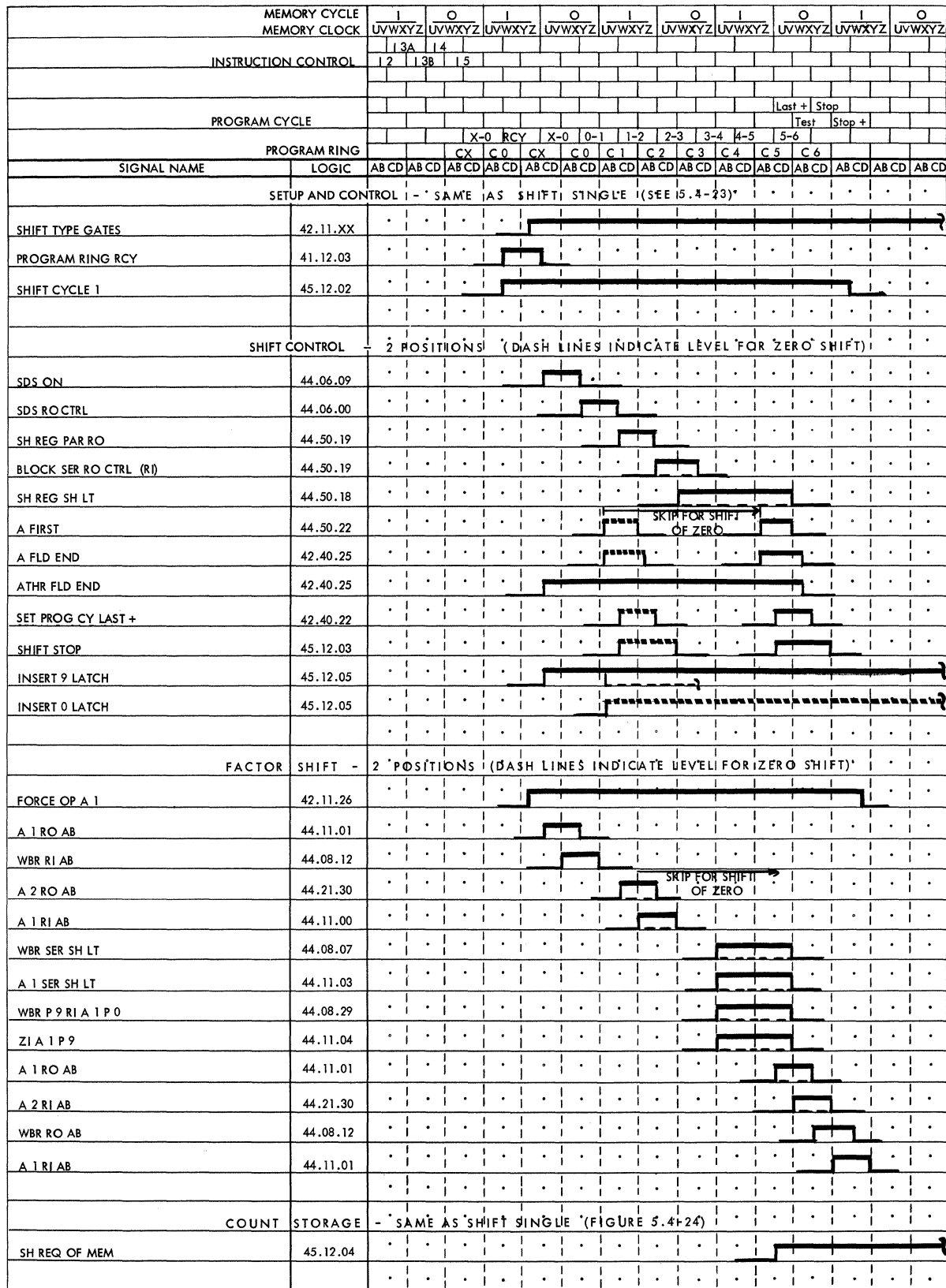


FIGURE 5.4-28A. SHIFT LEFT AND COUNT DOUBLE (A1 NON ZERO)

The shift register reads in from the significant digit scanner as a regeneration to show the correct zero count. To retain the value, the register is read out to the parallel output latches. On the following digit time, the value is read back into the register with a left shift to start the count-down. The output latches remain set to allow recording the shift value. The register continues the left shift until the bit reads out of the high-order position to set the A-first latch and the A-field end latch. If no shift is indicated by the immediate setting of the A-first latch, a zero is forced into the output latches. The program cycle ring is started at Last + to return the factor to the proper accumulators. The program ring is stopped in the normal manner at program cycle test.

At the end of cycle 1, the shift-request-of-memory latch is set to start the count storage cycle. This operation is identical to that for the shift-single operation. The index word at the address specified by the program field register is read into the auxiliary register. The shift values in the shift register and the tens digit insert latches are read out as complement values. They are fed through the adder with complement-add to enter the true count value in auxiliary register 2. The remainder of the index word is unchanged. The word is returned to its original location following the store request.

The program ring is started at data control 4 on the second cycle along with the setting of the shift cycle 2 latch. The CAB latches are not reset during the core storage transfers, thus the shift code type is maintained. The program cycle ring is not started for this cycle. The program ring stop is forced at C4. The store request is initiated at C3. The operation is ended at store control 3 by the development of the shift-code-end pulse.

Variations exist in the count value controls depending on the sense of zero conditions. Zeros must be inserted to provide for count storage and control. The tens digit insert must be adjusted due to adder carry with complement zeros. A count of zero requires a use of digit zero insert, which, when 9's complemented with an adder carry, reads out a zero. Counts of one through ten require a digit nine insert, while eleven through twenty require an eight. The latches are selected by a combination of significant digit scanner zero read-out and an A-first signal read-out at C1 indicating a shift of zero. The insert 9 latch is set at the start. If no shift is indicated, the insert 9 is reset, and the insert 0 is set. If accumulator 1 is all zeros, the insert 9 latch is reset, and the insert 8 latch is set. With a count of ten, the shift register bit reads out at C1 to reset the insert 8 latch, which was just set, and causes the insert 9 latch to be set again.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sh Cycle 2	DC4	2F-5F-45.12.02
Lt Dbl Sh Codes	CX	3G-42.11.29
Dbl 2 Cy Sh Codes	CX	2J-42.11.29
Sh Code Cy 1	Sh Cy 1	3C-42.11.32
Sh Lt Cnt	CX	3J-42.11.32

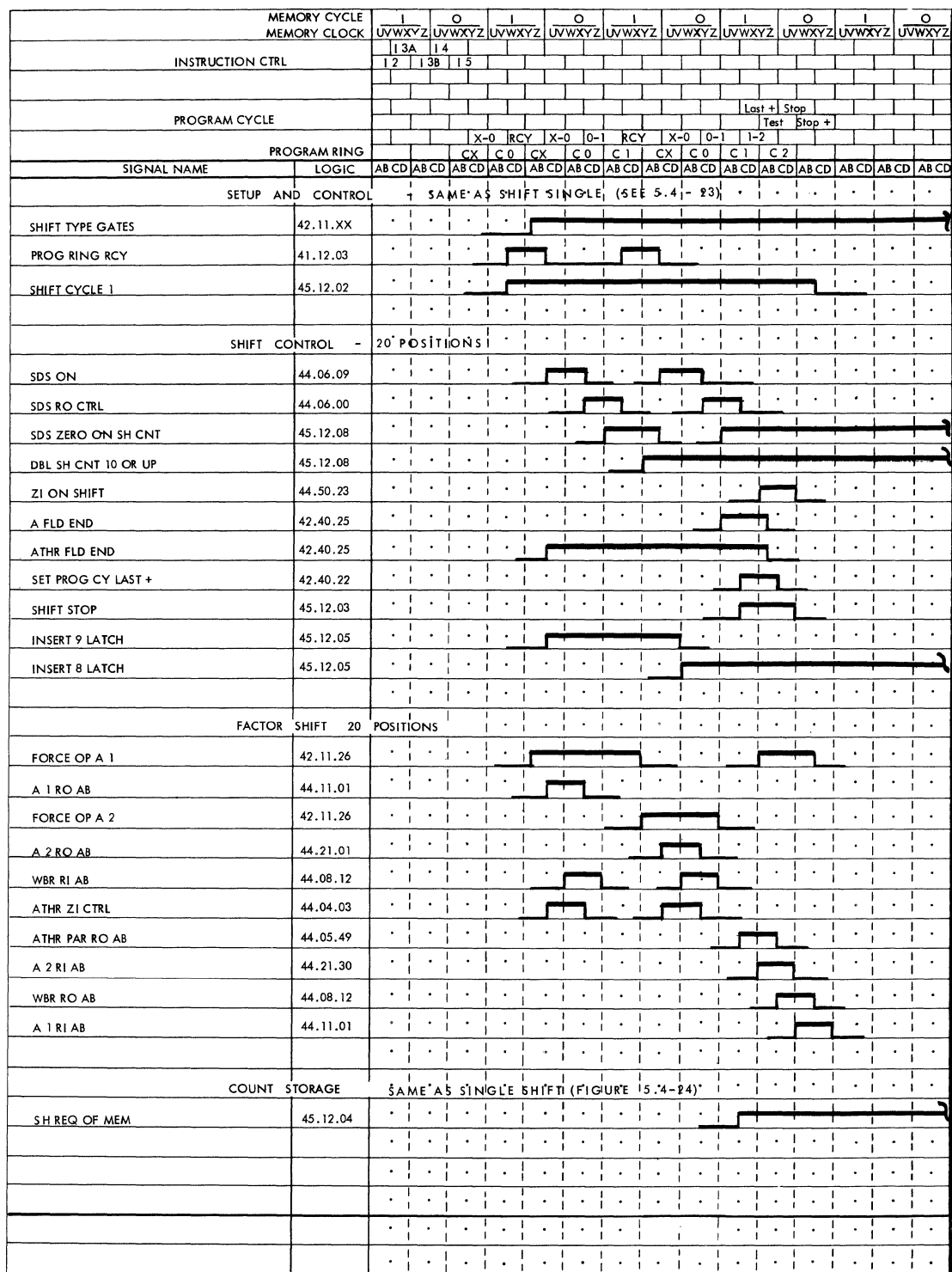


FIGURE 5.4-28C. SHIFT LEFT AND COUNT DOUBLE (A1 & A2 ZERO)

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh Lt Cnt Cy 1	Sh Cy 1	5H-42.11.32
Sh Lt Cnt Cy 2	Sh Cy 2	5G-42.11.32
2 Cy Sh Codes	CX	3G-42.11.33
Factor Shift, A 1 Non-Zero		
Force Op A 1	CX	3B-45.12.06
A 1 RO AB	CX-0	2D-44.11.01
SDS On	CX-0	2A-44.06.09
WBR RI AB Ctrl	CX-0	3A-44.08.12
A 2 RO AB	C1-2	3F-45.21.30
A 1 RI AB Ctrl	C1-2	4E-44.11.00
WBR Ser Sh Lt	C4	3H-4F-5H-44.08.07
A 1 Sh Lt L Turn On	C3-4	5G-44.11.02
WBR P 9 RI A 1 Psn 0		44.08.29
ZI A 1 Psn 9	C4	4E-44.11.04
A 1 RO AB	PC Last +	3E-44.11.01
A 2 RI AB Ctrl	PC Last +	3B-44.21.30
WBR AB RO	PC Stop	2F-44.08.12
A 1 AB RI Ctrl	PC Stop	3C-44.11.00
Factor Shift, A 1 Zero		
Force Op A 1	CX	3B-45.12.06
A 1 RO AB	CX-0	2D-44.11.01
SDS On	CX-0	2A-44.06.09
WBR RI AB Ctrl	CX-0	3A-44.08.12
SDS Zero On Sh Cnt	C1	4B-5B-45.12.08
Prg Rcy	SDS Z	2F-41.12.03

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Force Op A 2	Rey CX	3F-45.12.06
A 2 RO AB	CX-0	3A-44.21.00
SDS On	CX-0	2A-44.06.09
WBR RI AB Ctrl	CX-0	3A-44.08.12
WBR Ser Sh Lt	C4	3H-4F-5F-44.08.07
ZI WBR Psn 9	C4	2H-44.08.19
Reset WBR Lt Sh	Sh Stop AP	3J-4G-44.08.07
Force Op A 1	C2	3D-45.12.06
A 1 RO AB	PC Last +	3E-44.11.01
A 2 RI AB Ctrl	PC Last +	3B-44.21.30
WBR AB RO	PC Stop	2F-44.08.12
A 1 AB RI Ctrl	PC Stop	3C-44.11.00
Shift Register Operation		
Sh Reg Par Regen		3B-44.50.19
SDS Reg Zero	C1 AP	3B-44.06.05
SDS Zero On Sh Cnt	C1	4B-5B-45.12.08
Sh Reg Par RO Ctrl	C1-2	5C-44.50.19
ZI On Sh (0 Sh)	C1	5G-44.50.23
ZI On Sh (2 0 Sh)	C2	5H-44.50.23
Block Ser RO Ctrl (RI)	C2-3	2G-44.50.19
Sh Reg Sh Lt Ctrl	C2-3	3F-4F-5F-44.50.18
A-First	AP	3A-4B-5B-44.50.22
A Fld End Set	CX	4G-42.40.33
A Fld End (Normal)	A First	4E-5E-42.40.25
A Fld End (SDS Zero)	C1	3E-5E-42.40.25

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Set Prg Cy Last +	A Fld End	2F-4G-5G-42.40.22
Sh Stop	PC Last +	4B-45.12.03
Ins Nine on Sh Cnt (Basic)	CX	4B-5B-45.12.05
Ins Zero on Sh Cnt	C1	2G-5H-45.12.05
Ins Eight on Sh Cnt	C0	3E-5E-45.12.05
Ins Nine on Sh Cnt (10 Sh)	C1	2C-5B-45.12.05
Read-In Index Word		
Sh Req of Mem	PC Last +	4B-5B-45.12.04
Data Req	MOP	3H-4H-5G-42.40.04
Prevent CAB L Clear	PC Last +	4G-5G-42.11.24
Start Data Ctrl Ring	Addr Gate	3B-4B-5B-42.41.10
PR Fld Addr to CAB	DC1	4H-42.41.20
Prg Ring Start	DC4	4D-41.12.01
RI Auxr From IB	DC4	3D-5G-6G-42.31.18
Adder Operation		
RO Sh Reg OL Ch 2	CX	5F-44.50.23
Ins Tp Sh Cnt (9)	C0	5B-44.46.20
Ch 1 Ins 0	DC4	3G-4B-42.62.10
Turn On Comp Add	DC4	4D-44.41.14
Reset Comp Add	C1	4G-6G-44.41.17
Carry Insert	CX AP	3H-44.41.02
RO Adder	CX CP	3C-4B-5B-44.41.05
Adv Auxr 2	C0	4B-5E-42.31.11
Adv Stop (Auxr 2)	C4	5C-42.31.50
Auxr 2 RI Adder	C0	4B-5A-42.31.27

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Stp Auxr 2 RI Adder	C4	5B-6A-42. 31. 62
End Operation		
Store in Memory	C3	3F-4G-42. 41. 15
Store Req	MIP	2B-3B-42. 41. 14
Prg Ring Stop	C4 CP	3G-4G-5G-41. 12. 02
Adj Sign On Lt Sh	Prg Stp	4G-5G-45. 12. 15
Sh Code End	STC3	4G-45. 12. 03
End Data Op Mix	STC3	4F-42. 40. 30

5.4.03 -50 Shift Double - Shift from Point

The shift-from-point codes are a group of special shift-double functions used for adjust and edit operations. Accumulators 1 and 2 are used for the high and low factor values, respectively. Two of the four types specify a left and a right shift from a specified point digit in accumulator 1. The remaining two types perform the same functions for accumulator 2. The two shift types, calling for a shift away from the opposite register, are effectively single-shift operations. The portion of the factor in this register remains unchanged. In each of the four shift types the point digit may leave the specified register or be shifted within. This is dependent on the point location and the shift value. Zeros are inserted in the vacated area and digits shifting beyond the register limits are lost.

The setup operation for all types is the same. Up through the point of sensing the computer address bus latches, it resembles any other shift operation. The sensing of the hundreds position sets the controls for point operation and the direction of shift. The thousands position of the latches indicates the location of the point digit. The units and tens digit specify the amount of shift to be taken.

Two separate shift control systems, the field ring and the shift register, are used to count the shift values. The value of the point digit in the thousands position of the CAB latches is used to control the shift of the point digit to the edge of the register. When a right shift is specified, the field register is set "X9" to count from nine down through the point digit. For a left shift the register is set "0X" to count from the point digit down through zero. In either case the shift causes the point digit to just move out of the specified register. The same value is used a second time to move a portion of the digits back to their original position.

The shift register is set initially with the difference between the shift value and point value. This value, used to locate the point beyond the register, is computed by use of the adder. The point value is read from the field register, while the shift value is read directly from program register D. The normal read-out circuits are used for the field register. This requires the tens digit to feed channel 1 for a right shift, and

the units digit to feed channel 2 for a left shift. The data register right shifts to feed alternate channels under control of the shift direction. True-add is used for a right shift, because the point value is already the complement of the required shift. With a left shift, complement-add is used because the point value is equal to the required shift. Both the units and tens digits of the shift value are entered to obtain the full shift. Either a true or complement zero is inserted for the tens digit of the point value. The units position of the adder output is used to set the shift register. The zero non-zero detection is used to determine if the point remains within or shifts out of the specified register. A second signal is developed to indicate when the shift is greater than ten beyond the register. Count operations with the shift register are performed with a right shift to A-last +.

On shifts with the point remaining within the accumulator, the shift register is reset at the start of the second cycle. The units position of the shift value is read in from the CAB latches. This value is used to count the number of zeros to be inserted next to the point.

Each of the shift types follows a similar pattern. Two program cycles are required in all cases. The first cycle is used to shift the register so that the point split lays between two registers. The second cycle returns those digits which are not to be moved, inserts zeros in the amount of the shift value, and places the point digit and remainder of digits beyond the zeros. The second cycle varies between point within and point shifted out of the register. Because of register limitations, in their ability to shift, each type uses a different grouping of registers. This entails parallel transfers between registers over the arithmetic bus. These may occur at the start of cycle one, during the program cycle control, the start of cycle two, or the end of the shifting.

The following signals are developed during the setup portion of the operation. They include signals required to determine the type of shift and the movement of the point (Figure 5.4-29).

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Control Setup		
Prg Ring Rcy Ctrl	C0	2F-41.12.03
RO PR Data for Sh	Prg Rcy	3B-4B-45.12.01
PR Data RO CAB	Prg Rcy	5G-42.02.04
Set CAB L Reg	CX AP	4H-5F-42.05.01
Sh Cycle 1	Prg Rcy	4B-5B-45.12.02
Sh Cycle 2	PC Stop	2G-5F-45.12.02
Dbl Sh Code	CX	2A-42.11.25
Sh Code	CX	2A-42.11.32
Sh RT PT A 1	CX	4E-42.11.25

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh RT PT A 2	CX	4G-42.11.25
Sh LT PT A 1	CX	4F-42.11.25
Sh LT PT A 2	CX	4H-42.11.25
Sh From PT	CX	5F-42.11.25
Sh RT PT Codes	CX	3A-42.11.36
Sh LT PT Codes	CX	3C-42.11.36
RT Dlb Sh Codes	CX	5J-42.11.29
LT Dbl Sh Codes	CX	2G-42.11.29
Sh RT PT Cy 1 Codes	Sh Cy 1	4B-42.11.36
Sh LT PT Cy 1 Codes	Sh Cy 1	4D-42.11.36
Field Register Set		
Fld Reg U RI CAB L Th	C0	5A-42.07.02
Fld Reg T RI CAB L Th	C0	5C-42.07.02
Ins Zero Fld Reg TP	C0	5E-42.07.02
Ins Dec 9 UP Fld Reg	C0	5H-42.07.02
Adder Entry - Shift Right		
RO Fld Reg TP Ch 1	C0	3C-42.07.01
PR Data Ser Adv	C0	4H-3E-5D-42.02.02
Turn On PR RO Ch 2	C0	4F-42.02.08
Turn Off PR RO Ch 2	C2	4H-42.02.08
Ch 1 Ins 0	C1	6F-42.62.10
True Add		
No-Carry Ins	CX-0	3C-44.41.02
Adder Entry - Shift Left		
RO Fld Reg Up Ch 2	C0	3E-42.07.01

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
PR Data Ser Adv	C0	4H-3E-5D-42.02.02
Turn On PR RO Ch 1	C0	4B-42.02.08
Turn Off PR RO Ch 1	C2	4D-42.02.08
Ins 9 Ch 2 (3B)	C1	6H-44.46.20
Comp Add		
No-Carry Ins	CX-0	3C-44.41.02
Adder Read-Out		
RO Adder	C0 CP	4A-5B-44.41.05
Sh Reg RI Adder	C1	3G-44.50.07
Reset Zero Non-Zero	C1	4H-44.41.17
Adder Zero	C1	3B-4B-44.40.20
Adder Non-Zero (0B)	C2 AP	2E-4E-44.40.20
Sense Adder Output	AP	5B-44.40.18
Sh PT Out A 1 & A 2	C2 CP	4E-5E-45.12.16

Shift Right - Point in Accumulator 1 (X4XX) (Figure 5.4-30)

The shift right from point in accumulator 1 code calls for a split in the accumulator 1 word to the left of the specified point digit. The point digit and digits to the right in accumulator 1 and all digits in accumulator 2 are right shifted by the specified shift value.

This code requires the use of an abnormal shift procedure. Normal procedure would require the use of three registers capable of left shift. To overcome the problem, the portion of the factor, which right shifts out of accumulator 2, is reversed by entering the factor into the low order of the word buffer register with a left shift. During the second cycle, when the factor is returned, the process is repeated with the factors interchanged so that the last digit shifted out is the first to be replaced.

At the start of the first cycle no parallel transfers are required to position the factors. Accumulators 1 and 2 are right shifted, while the word buffer register shifts left. The low order of accumulator 1 feeds the high order of accumulator 2. The low order of accumulator 2 feeds the low order of the word buffer register to reverse the factor.

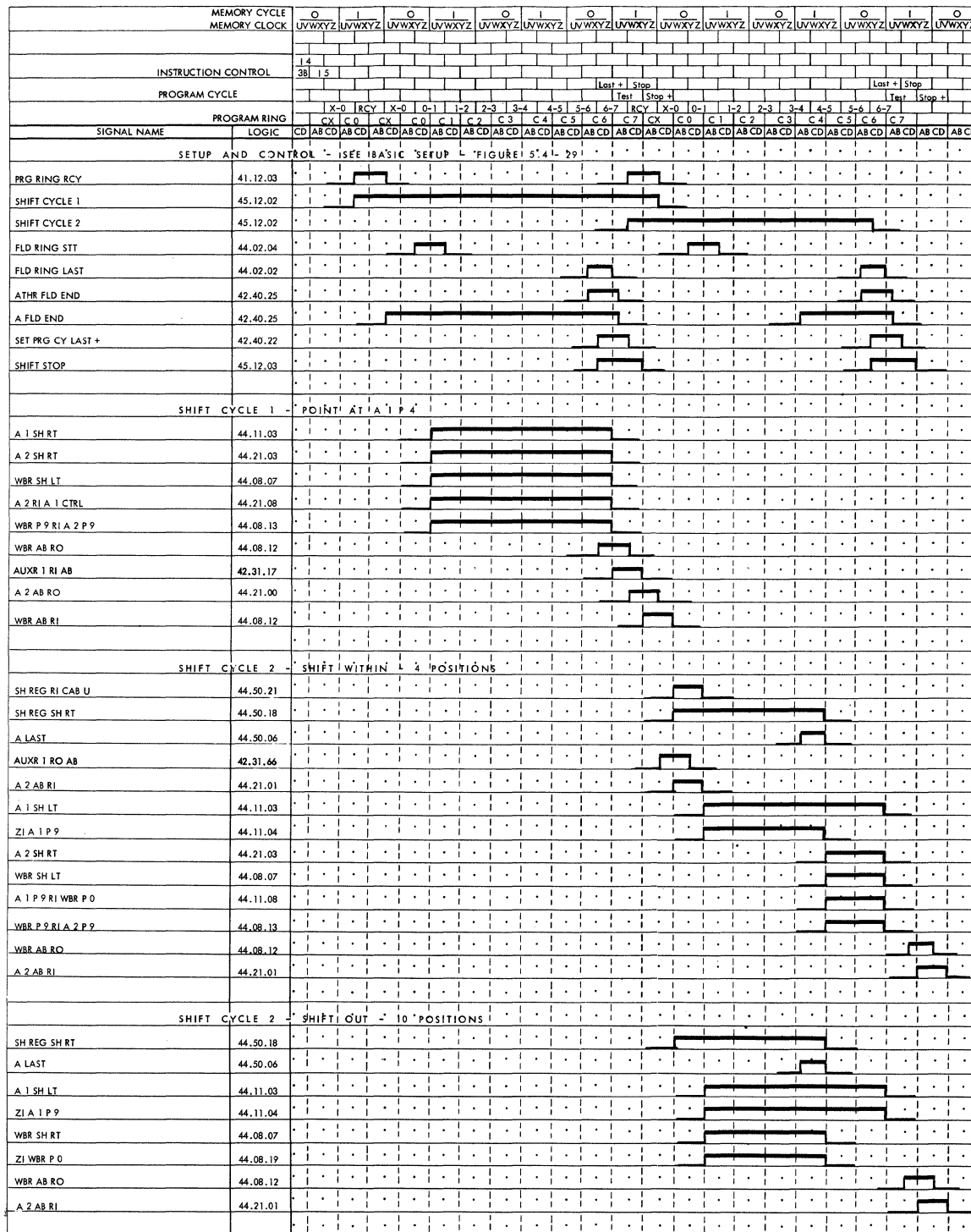


FIGURE 5.4-30. SHIFT RIGHT POINT IN AT

The shift stops with the setting of Program Cycle Last + by Field Ring Last through setting Athr Field End. The A-field end is forced. During the program cycle, the word buffer register is transferred to the auxiliary register, and accumulator 2 is transferred to the word buffer register. The operation of the first cycle is the same for both a shift of point within and shift out.

Point Within. When adder output zero is developed in the setup, gates are set to shift the point digit back into accumulator 1. The shift register is reset, and the shift value from the units position of the CAB latches is read in. Before the shift, the auxiliary register is transferred to accumulator 2. This completes an interchange between accumulator 2 and the word buffer register. To start the shift, accumulator 1 is left shifted under control of the field ring set with the point value. Zeros are inserted into the low order of accumulator 1 under control of the shift register. When the A-last point is reached, the zero insert is stopped, and the digits from the high order of the word buffer register are shifted in until field-ring-last. The word buffer register is left shifted and accumulator 2 is right shifted for this portion of the operation. The digits from the low order of accumulator 2 are fed to the low order of the word buffer register to enter the reversed digits in their original order. The shifting is stopped by setting Athr Field End, Prg Cy Last +, and Shift Stop in sequence. At program cycle stop the word buffer register is transferred back to the accumulator 2 to complete the shift.

Point Shifts Out. An adder-output non-zero during the setup develops gates to move the point further to the right. For this operation accumulator 2 and the reversed factor are not required. Accumulator 1 is left shifted under control of the field ring with the point value. Zeros are inserted at the low order until Field Ring Last +. Simultaneously, the word buffer register is right shifted under control of the shift register using the computed value. Zeros are inserted at the high order until A-last +. Field Ring Last sets Athr Field End, and A-last sets A-field end to set Program Cycle Last +. At program cycle stop the word buffer register is transferred back to accumulator 2 to complete the shift.

The shift-end pulse is developed at program cycle stop for either type of shift to end the operation. The sign of accumulator 2 is set to that of accumulator 1 in the same manner as for shift-right-double operations.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh from PT Cy 1	Sh Cy 1	5F-42.11.36
Sh RT PT A 1 Cy 1	Sh Cy 1	3G-42.11.36
Sh from PT Cy 2	Sh Cy 2	3F-42.11.37
Sh RT PT A 1 Cy 2	Sh Cy 2	3G-42.11.37
Sh PT Within A	Sh Cy 2	4D-45.12.11
Sh PT Within A 1 RT	Sh Cy 2	3C-45.12.12
Sh PT Out of A	Sh Cy 2	4H-45.12.11

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Sh PT Out of A 1 RT	Sh Cy 2	3G-45.12.12
Adj Sign On RT Sh	Prg RG Stop	4B-5B-45.15.15
Shift Cycle 1		
A 1 Sh RT L Turn On	C0-1	3A-44.11.12
A 2 Sh RT L Turn On	C0-1	4D-44.21.02
WBR Sh Lt Turn On	C0-1	2E-44.08.37
A 2 RI A 1 Ctrl	C1	5D-44.21.08
WBR P 9 RI A 2 P 9	C1	5E-44.08.13
Fld Ring Stt Pulse	C0 CP	3F-5A-44.02.04
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02
A Fld End	CX-0	4H-42.40.33
Athr Fld End	FR Last	4B-5B-42.40.25
Set Prg Cy Last +	CP	2F-4G-5G-42.40.22
Sh Stop	PC Last +	4B-5B-45.12.03
A 1 Sh RT L Reset	Sh Stop	2E-44.11.02
A 2 Sh RT L Reset	Sh Stop	4G-44.21.02
WBR Sh Lt Turn Off	Sh Stop	3J-44.08.07
WBR AB RO Ctrl	PC Last +	3E-44.08.15
Auxr 1 RI AB Ctrl	PC Last +	3A-42.31.17
A 2 AB RO Ctrl	PC Stop	3E-44.21.30
WBR RI AB Ctrl	PC Stop	3B-44.08.12
Shift Cycle 2 - Shift Within		
Sh Reg RI CAB U	CX-0	3B-44.50.21
Sh Reg Sh RT Ctrl On	CX-0	2C-44.50.18
Auxr 1 RO AB Ctrl	CX-0	4H-42.31.66

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 2 AB RO Ctrl	CX-0	3D-44. 21. 30
A 1 Sh Lt L Turn On	C0-1	4F-44. 11. 12
Z Ins A 1 9 P	C1	3F-4F-44. 11. 04
Fld Ring Stt Pulse	C0	3F-5A-44. 02. 04
A-Last	AP	3B-4B-5B-44. 50. 06
Reset ZI A 1	A-Last +	2G-3G-44. 11. 04
A 2 Sh RT L Turn On	A-Last +	4E-44. 21. 02
WBR Sh Lt Turn On	A-Last +	6E-44. 08. 37
A 1 P 9 RI WBR P 0	A-Last +	3F-5G-44. 11. 08
WBR P 9 RI A 2 P 9	A-Last + AP	5D-44. 08. 13
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02
A Fld End	A-Last	4E-5E-42. 40. 25
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-5B-45. 12. 03
A 1 Sh Lt L Reset	Sh Stop	5J-44. 11. 02
A 2 Sh RT L Reset	Sh Stop	4G-44. 21. 02
WBR Sh Lt Turn Off	Sh Stop	3J-44. 08. 07
WBR AB RO	PC Stop	3J-44. 08. 15
A 2 AB RI Ctrl	PC Stop	3C-44. 21. 30
Shift Cycle 2 - Shift Out		
A 1 Sh Lt L Turn On	C0-1	4F-44. 11. 12
ZI A 1 9 P	C1	5G-44. 11. 04
Fld Ring Stt Pulse	C0	3F-5A-44. 02. 04
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 1 Sh Lt L Reset	F R Last +	4H-44.11.12
WBR Sh RT Turn On	C0-1	4F-44.08.37
Z Ins WBR 0 P	C1	3G-44.08.19
Sh Reg Sh RT Ctrl On	CX-0	2D-44.50.18
A-Last	AP	3B-4B-5B-44.50.06
WBR Sh RT Turn Off	A-Last +	4G-44.08.37
A Fld End	A-Last	4E-5E-42.40.25
Athr Fld End	F R Last	4B-5B-42.40.25
Set Prg Cy Last +	CP	2F-4G-5G-42.40.22
WBR AB RO	PC Stop	3J-44.08.15
A 2 AB RI Ctrl	PC Stop	3C-44.21.30
End Operation		
Sh Code End	PC Stop	5H-45.12.03
End Data Op Mix	PC Stop	4F-42.40.30

Shift Right - Point in Accumulator 2 (X5XX) (Figure 5.4-31)

The shift right from point in accumulator 2 code calls for a split in the accumulator 2 word to the left of the specified point digit. The point digit and digits to the right in accumulator 2 are shifted to the right by the specified shift value. The digits in accumulator 1 are not changed but are moved to the auxiliary register during the operation to allow the use of accumulator 1 in the shift.

At the start of the first cycle accumulator 2 is transferred to the word buffer register for shifting, and accumulator 1 is transferred to the auxiliary register for storage. The factor is right shifted in the word buffer register feeding into accumulator 1 by right shift. The shift continues under control of the field ring until field-ring-last. This signal sets the Arith Field End, which along with a forced A-field end, set the program cycle at Last +. No parallel transfers are required between cycles.

Point Within. When adder output zero is developed in the setup, gates are set to shift the point digit back into accumulator 2. The shift register is reset, and the shift value from the units position of the CAB latches is read in. To start the shift, the word buffer register is left shifted under control of the field ring set with the point value.

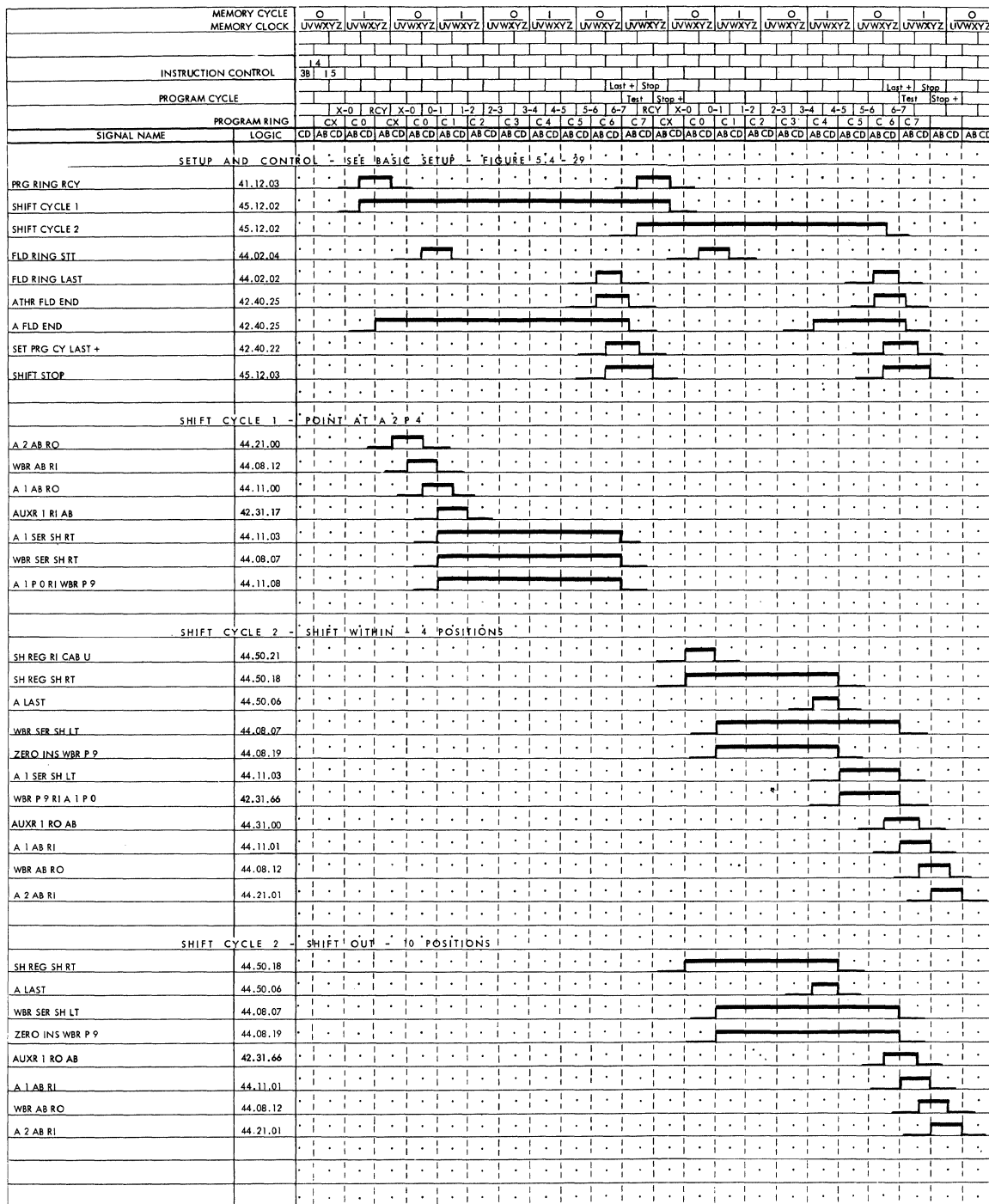


FIGURE 5.4-31. SHIFT RIGHT POINT IN A2

Zeros are inserted into the low order of the word buffer register under control of the shift register. When the A-last point is reached, the zero insert is stopped, and the digits from the high order of accumulator 1 are shifted in until field ring last. Accumulator 1 is left shifted for this portion of the operation. The shifting is stopped by setting Athr Field End, Program Cycle Last +, and Shift Stop in sequence. At program cycle last +, the auxiliary register is transferred to accumulator 1, and at program cycle stop-the-word buffer register is returned to accumulator 2 to end the shift.

Point Shifts Out. An adder-output non-zero during the setup develops gates to return the now low-order digits in accumulator 2 to their original location. The point digit and low-order digits are lost. For this portion of the operation, the word buffer register with the accumulator 2 factor is left shifted under control of the field ring. Zeros are inserted at the low order until field ring last. The shift register is not used but is shifted to set the A-last latch. These signals set Athr Field End and A Field End, respectively, which sets Program Cycle Last +. At Program Cycle Last +, the auxiliary register is transferred to accumulator 1, and at program cycle stop, the word buffer register is returned to accumulator 2 to end the shift.

The shift-end pulse is developed at program cycle stop for either type of shift to end the operation. The signs are not changed.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh from PT Cy 1	Sh Cy 1	5F-42.11.36
Sh RT PT A 2 Cy 1	Sh Cy 1	3H-42.11.36
Sh from PT Cy 2	Sh Cy 2	3F-42.11.37
Sh RT PT A 2 Cy 2	Sh Cy 2	3H-42.11.37
Sh PT within A	Sh Cy 2	4D-45.12.11
Sh PT within A 2 RT	Sh Cy 2	3B-45.12.12
Sh PT Out of A	Sh Cy 2	4H-45.12.11
Sh PT Out of A 2 RT	Sh Cy 2	3H-45.12.12
Shift Cycle 1		
A 2 AB RO Ctrl	CX-0	2H-44.21.30
WBR AB RI Ctrl	CX-0	2C-44.08.12
A 1 AB RO Ctrl	C0-1	3G-44.11.17
Auxr 1 RI AB Ctrl	C0-1	3B-42.31.17
A 1 Sh RT L Turn On	C0-1	3A-44.11.12
WBR Sh RT Turn on	C0-1	3A-44.08.37
A 1 P 0 RI WBR P 9	C1	3B-44.11.08
Fld Ring Stt Pulse	C0 CP	3F-5A-44.02.04
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02
A Fld End	CX-0	4H-42.40.33

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-5B-45. 12. 03
A 1 Sh RT L Reset	Sh Stop	2E-44. 11. 02
WBR Sh R Turn Off	Sh Stop	3B-44. 08. 07
Shift Cycle 2 - Shift Within		
WBR Sh Lt Turn On	C0-1	5E-44. 08. 37
Z Ins WBR P 9	C1	4F-44. 08. 19
Fld Ring Stt Pulse	C0 CP	3F-5A-44. 02. 04
Sh Reg RI CAB U	CX-0	3B-44. 50. 21
Sh Reg Sh RT Ctrl On	CX-0	2C-44. 50. 18
A-Last	AP	3B-4B-5B-44. 50. 06
Reset Z Ins WBR	A-Last +	3F-4G-44. 08. 19
A 1 Sh Lt L Turn On	A-Last +	4E-44. 11. 12
WBR P 9 RI A 1 P 0	A-Last +	4F-44. 08. 29
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02
A Fld End	A-Last	4E-5E-42. 40. 25
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-5B-45. 12. 03
A 1 Sh Lt L Reset	Sh Stop	5J-44. 11. 02
WBR Sh Lt Turn Off	Sh Stop	4J-42. 31. 66
Auxr 1 RO AB Ctrl	PC Last +	3G-44. 31. 10
A 1 AB RI Ctrl	PC Last +	3B-44. 11. 17
WBR AB RO Ctrl	PC Stop	3F-44. 08. 15

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 2 AB RI Ctrl	PC Stop	2C-44. 21. 30
Shift Cycle 2 - Shift Out		
Sh Reg Sh RT Ctrl On	CX-0	2D-44. 50. 18
WBR Sh Lt Turn On	C0-1	5E-44. 08. 37
Z Ins WBR P 9	C1	4F-44. 08. 19
Fld Ring Stt Pulse	C0 CP	3F-5A-44. 02. 04
A-Last	AP	3B-4B-5B-44. 50. 06
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02
A Fld End	A-Last	4E-5E-42. 40. 25
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
WBR Sh Lt Turn Off	FR Last +	5H-44. 08. 39
Auxr 1 RO AB Ctrl	PC Last +	4J-42. 31. 66
A 1 AB RI Ctrl	PC Stop	3F-44. 08. 15
WBR AB RO Ctrl	PC Stop	3F-44. 08. 15
A 2 AB RI Ctrl	PC Stop	2C-44. 21. 30
End Operation		
Sh Code End	PC Stop	5H-45. 12. 03
End Data Op Mix	PC Stop	4F-42. 40. 30

Shift Left - Point in Accumulator 1 (X6XX) (Figure 5.4-32)

The shift left from point in accumulator 1 code calls for a split in the accumulator 1 word to the right of the specified point digit. The point digit and the digits to the right in accumulator 1 are shifted to the left by the specified shift value. The digits in accumulator 2 are not moved or changed during the operation.

At the start of the first cycle, accumulator 1 is transferred to the word buffer register for the shift operation. Both accumulator 1 and the word buffer register are left shifted under control of the field ring. The high order of the word buffer register feeds the low order of accumulator 1 with the overflow digits. The shift stops at field ring

digits from the low order of accumulator 1 are shifted in until field ring last. Accumulator 1 is right shifted during this portion of the operation. The shifting is stopped by setting Athr Field End, Program Cycle Last +, and Shift Stop in sequence. At program cycle stop, the word buffer register is transferred to accumulator 1 to end the shift.

Point Shifts Out. An adder-output non-zero during the setup develops gates to return the now high order digits in accumulator 1 to their original location. The point digit and low order digits are lost. For this portion of the operation, the word buffer register with the accumulator 1 factor is right shifted under control of the field ring. Zeros are inserted at the high order until field ring last. The shift register is not used but is shifted to set the A-last latch. These signals set Athr Field End and A Field End, respectively, which sets Program Cycle Last +. At program cycle stop, the word buffer register is transferred to accumulator 1 to end the shift.

The shift-end pulse is developed at program cycle stop for either type of shift to end the operation. The signs are not changed.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh from PT Cy 1	Sh Cy 1	5F-42.11.36
Sh Lt PT Ins 1	C1	4C-42.11.36
Sh from PT Cy 2	Sh Cy 2	3F-42.11.37
Sh Lt PT A 1 Cy 2	Sh Cy 2	3D-42.11.37
Sh PT within A	Sh Cy 2	4D-45.12.11
Sh PT within A 1 Lt	Sh Cy 2	4C-45.12.11
Sh PT Out of A	Sh Cy 2	4H-45.12.11
Sh PT Out of A 1 Lt	Sh Cy 2	4J-45.12.11
Shift Cycle 1		
A 1 AB RO Ctrl	CX-0	3F-44.11.17
WBR AB RI Ctrl	CX-0	3D-44.08.12
WBR Sh Lt Turn On	C0-1	3E-44.08.37
A 1 Sh Lt L Turn On	C0-1	3E-44.11.12
A 1 P 9 RI WBR P 0	C1	6F-44.11.08
Fld Ring Stt Pulse	C0 CP	3F-5A-44.02.04
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A Fld End	CX-0	4H-42. 40. 33
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-5B-45. 12. 03
WBR Sh Lt Turn Off	Sh Stop	3J-44. 08. 07
A 1 Sh Lt L Reset	Sh Stop	5J-44. 11. 02
Shift Cycle 2 - Shift Within		
Sh Reg RI CAB U	CX-0	3B-44. 50. 21
Sh Reg Sh RT Ctrl On	CX-0	2C-44. 50. 18
WBR Sh RT Turn On	C0-1	4E-44. 08. 37
Z Ins WBR P 0	C1	4C-5C-44. 08. 19
Fld Ring Stt Pulse	C0	3F-5A-44. 02. 04
A-Last	A	3B-4B-5B-44. 50. 06
Reset Z Ins WBR	A-Last +	3D-44. 08. 19
A 1 Sh RT L Turn On	A-Last +	4B-44. 11. 12
WBR P 0 RI A 1 P 9	A-Last +	2D-44. 08. 29
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02
A Fld End	A-Last	4E-5E-42. 40. 25
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
Sh Stop	PC Last +	4B-5B-45. 12. 03
WBR Sh RT Turn Off	Sh Stop	3B-4C-44. 08. 07
A 1 Sh RT L Reset	Sh Stop	2E-44. 11. 02
WBR AB RO Ctrl	PC Stop	3G-44. 08. 15
A 1 AB RI Ctrl	PC Stop	3A-44. 11. 17

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Shift Cycle 2 - Shift Out		
Sh Reg Sh RT Ctrl On	CX-0	2D-44. 50. 18
WBR Sh RT Turn On	C0-1	4E-44. 08. 37
Z Ins WBR P 0	C1	2B-44. 08. 19
Fld Ring Stt Pulse	C0	3F-5A-44. 02. 04
Fld Ring Last	Fld Stp Match	3A-4A-44. 02. 02
A-Last	AP	3B-4B-5B-44. 50. 06
A Fld End	A-Last	4E-5E-42. 40. 25
Athr Fld End	FR Last	4B-5B-42. 40. 25
Set Prg Cy Last +	CP	2F-4G-5G-42. 40. 22
WBR Sh RT Turn Off	FR Last +	5F-44. 08. 39
WBR AB RO Ctrl	PC Stop	3G-44. 08. 15
A 1 AB RI Ctrl	PC Stop	3A-44. 11. 17
End Operation		
Sh Code End	PC Stop	5H-45. 12. 03
End Data Op Mix	PC Stop	4F-42. 40. 30

Shift Left - Point in Accumulator 2 (X7XX) (Figure 5.4-33)

The shift left from point in accumulator 2 code calls for a split in the accumulator 2 word to the right of the specified point digit. The point digit and digits to the left in accumulator 2 and all digits in accumulator 1 are left shifted by the specified shift value.

At the start of the first cycle accumulator 2 is transferred to the word buffer register, where it can be left shifted. Both accumulator 1 and the word buffer register are left shifted with the high order position of each feeding the low order of the other in a closed ring. The shift is controlled by the field ring with the point value. The shift stops at field ring last by setting Athr Field End, Program Cycle Last +, and Shift Stop in sequence. At program cycle stop, the word buffer register reads out its factor to accumulator 2 and regenerates. The additional register is needed as zeros are entered.

Point Within. When adder output zero is developed in the setup, gates are set to shift the point digit back into accumulator 2. The shift register is reset and the shift value from the units position of the CAB latches is read in. To start the shift, accumulator 2

is right shifted under control of the field ring set with the point value. Zeros are inserted into the high order of accumulator 2 under control of the shift register. When the A-last point is reached, the zero insert is stopped and the digits from the low order of accumulator 1 are shifted in until field ring last. The word buffer register is also right shifted to feed its low-order digits into the high order of accumulator 1. The shifting is stopped by setting Athr Field End, Program Cycle Last +, and Shift Stop in sequence. No parallel transfers are required to end the shift operation because the final factor is properly positioned in accumulators 1 and 2.

Points Shifts Out. An adder-output non-zero during the setup develops gates to move the point further to the left. For this operation the word buffer register is not required. Accumulator 2 is right shifted under control of the field ring with the point value. Zeros are inserted at the low order until Field Ring Last +. Simultaneously, accumulator 1 is left shifted under control of the shift register using the computed value. Zeros are inserted at the low order until A-last +. The field ring last sets the A-field end to set Program Cycle Last +. No parallel transfers are required to end the shift operation because the final factor is properly positioned in accumulators 1 and 2.

The shift-end pulse is developed at program cycle stop for either type of shift to end the operation. The sign of accumulator 1 is set to that of accumulator 2 in the same manner as for shift-left-double operations.

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
Operation Code Gates		
Sh from PT Cy 1	Sh Cy 1	5F-42.11.36
Sh Lt PT Ins 1	C1	4C-42.11.36
Sh from PT Cy 2	Sh Cy 2	3F-42.11.37
Sh Lt PT A 2 Cy 2	Sh Cy 2	3E-42.11.37
Sh PT within A	Sh Cy 2	4D-45.12.11
Sh PT within A 2 Lt	Sh Cy 2	4B-45.12.11
Sh PT Out of A	Sh Cy 2	4H-45.12.11
Sh PT Out of A 2 Lt	Sh Cy 2	4G-45.12.11
Adj Sign On Lt Sh	PR Stop	4G-5G-45.12.15
Shift Cycle 1		
A 2 AB RO Ctrl	CX-0	3H-44.21.30
WBR AB RI Ctrl	CX-0	2B-44.08.12
A 1 Sh Lt L Turn On	C0-1	3E-44.11.12
WBR Sh Lt Turn On	C0-1	3E-44.08.37

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A 1 P 9 RI WBR P 0	C1	6F-44.11. 08
WBR P 9 RI A 1 P 0	C1	3F-44.08.29
Fld Ring Stt Pulse	C0	3F-5A-44.02.04
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02
A Fld End	CX-0	4H-42.40.33
Athr Fld End	FR Last	4B-5B-42.40.25
Set Prg Cy Last +	CP	2F-4G-5G-42.40.22
Sh Stop		45.12.03
A 1 Sh Lt L Reset	Sh Stop	5J-44.11.02
WBR Sh Lt Turn Off	Sh Stop	3J-44.08.07
WBR AB RO Ctrl	PC Stop	3H-44.08.15
A 2 AB RI Ctrl	PC Stop	2D-44.21.30
Shift Cycle 2 - Shift Within		
Sh Reg RI CAB U	CX-0	3B-44.50.21
Sh Reg Sh RT Ctrl On	CX-0	2C-44.50.18
A 2 Sh RT L Turn On	C0-1	2A-44.21.02
Z Ins A 2 P 0	C0-1	4E-44.21.04
Fld Ring Stt Pulse	C0	3F-5A-44.02.04
A-Last	AP	3B-4B-5B-44.50.06
Reset Z Ins A 2	A-Last +	3F-4F-44.21.04
A 1 Sh RT L Turn On	A-Last +	4B-44.11.12
WBR Sh RT Turn On	A-Last +	4C-44.08.37
A 2 RI A 1 Ctrl	A-Last +	4A-4B-5B-44.21.08
A 1 P 0 RI WBR P 9	A-Last +	4A-44.11.08
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02

<u>Major Signals</u>	<u>Key Timing</u>	<u>Logic and Location</u>
A Fld End	A-Last	4E-5E-42.40.25
Athr Fld End	FR Last	4B-5B-42.40.25
Set Prg Cy Last +	CP	2F-4G-5G-42.40.22
Sh Stop	PC Last +	4B-5B-45.12.03
A 1 Sh RT L Reset	Sh Stop	2E-44.11.02
A 2 Sh RT L Reset	Sh Stop	4G-44.21.02
WBR Sh RT Turn Off	Sh Stop	3B-44.08.07
Shift Cycle 2 - Shift Out		
A 2 Sh RT L Turn On	C0-1	2A-44.21.02
Z Ins A 2 P 0	C0-1	4E-44.21.04
Fld Ring Stt Pulse	C0	3F-5A-44.02.04
Fld Ring Last	Fld Stp Match	3A-4A-44.02.02
A 2 Sh RT L Reset	FR Last +	4H-44.21.02
A 1 Sh Lt L Turn On	C0-1	3G-44.11.12
Z Ins A 1 P 9	C1	5H-44.11.04
Sh Reg Sh RT Ctrl On	CX-0	2D-44.50.18
A-Last	AP	3B-4B-5B-44.50.06
A 1 Sh Lt L Reset	A-Last +	4J-44.11.12
A Fld End	A-Last	4E-5E-42.40.25
Athr Fld End	FR Last	4B-5B-42.40.25
Set Prg Cy Last +	CP	2F-4G-5G-42.40.22
End Operation		
Sh Code End	PC Stop	5H-45.12.03
End Data Op Mix	PC Stop	4F-42.40.30

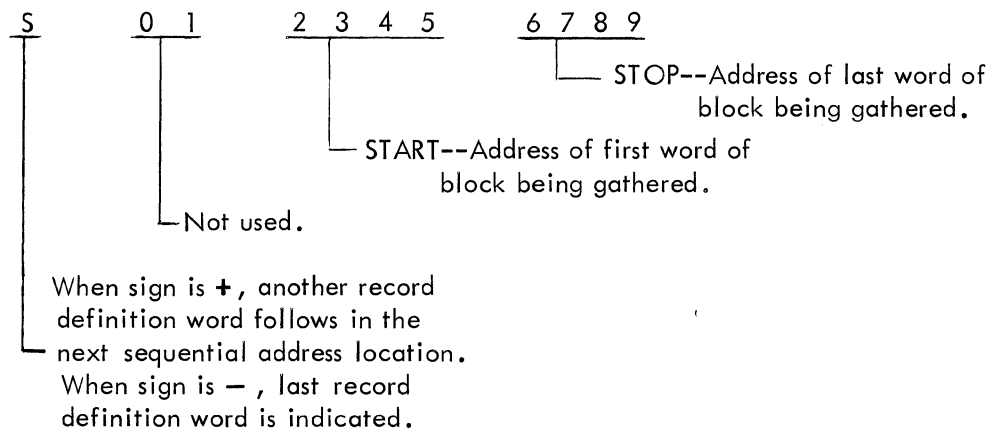
5. 5. 00 BLOCK TRANSFER INSTRUCTIONS

5. 5. 01 Record Scatter and Record Gather

Logical circuits and machine timing for these two operation codes are substantially the same. Record Gather is described in detail followed by a description of the minor differences involved in the record scatter operation.

Op -65 Record Gather (RG)

The gather operation moves scattered blocks of data to a single block of sequential core storage locations. The first word of the single block of core storage locations is defined by the indexing portion of the index word specified by the address in positions 4 and 5 of the instruction. The locations of the scattered blocks of data to be gathered are specified by Record Definition Words (also called Record Control Word, RCW). The location of the first Record Definition Word (RDW) is specified by positions 6 - 9 of the instruction. The format of the record definition word is as follows:



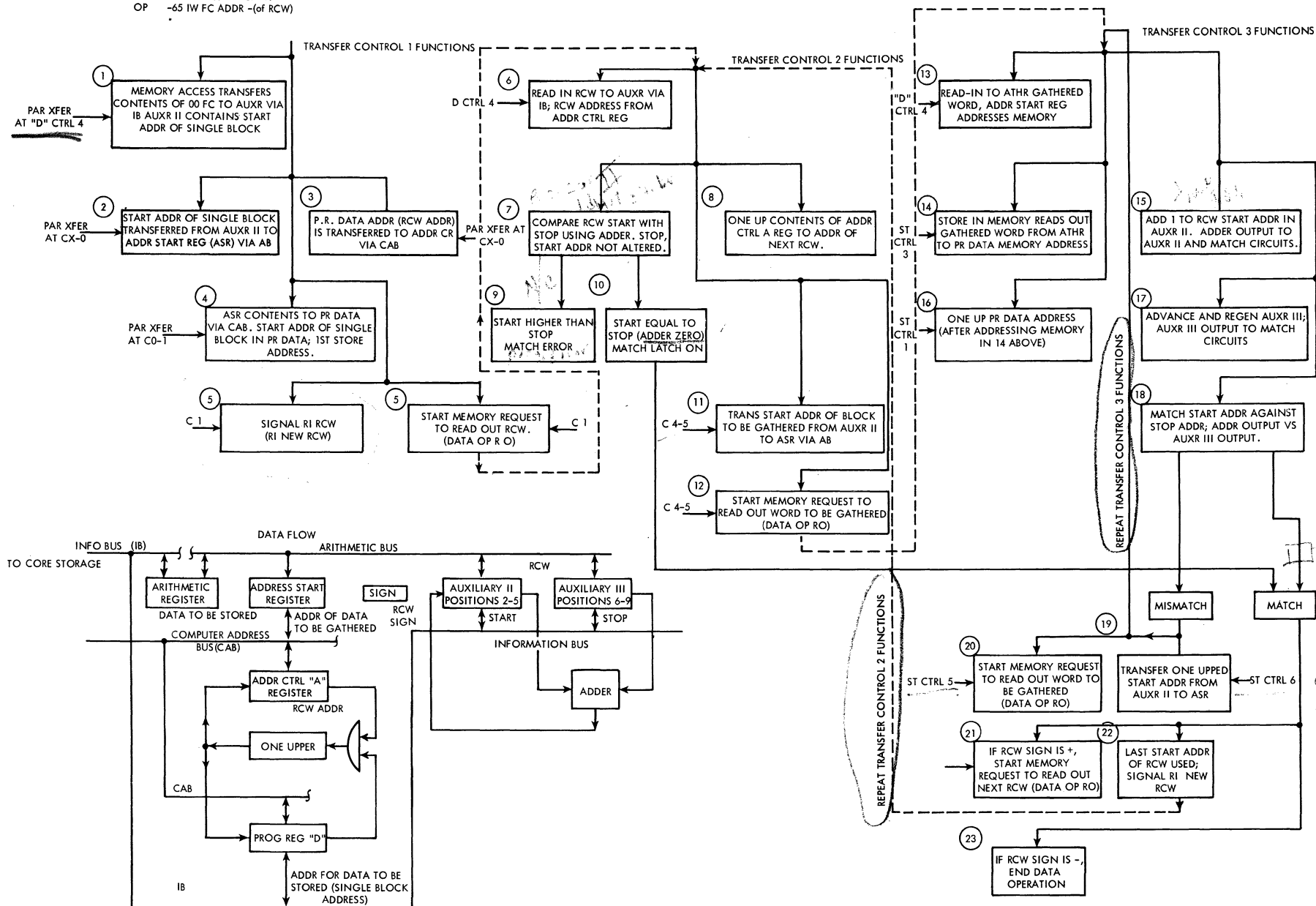
If only one block of data is to be gathered, the sign of the record definition word must be minus. Record definition words must be in sequential core storage locations.

Op +65 Record Scatter (RS)

The scatter operation moves blocks of data from a single block of sequential storage locations to the scattered blocks specified by the record definition word (or words). The first word of the single block is defined by the indexing portion of the index word specified by the address in positions 4 and 5 of the instruction. The locations of the scattered blocks of data are defined by record definition words in the same manner as with the record gather operation. The first record definition word is in the location specified by positions 6 - 9 of the instruction.

S 01 23 45 67 89
OP -65 IW FC ADDR -(of RCW)

FIGURE 5.5-2. BLOCK TRANSMISSION-RECORD GATHER



Self. Out 3/00.70

Kenn. 3/00.70

- ③ The address in Program Register D is the address of the first record control word. It is moved from the PR Data register to its permanent location in the Address Control A Register.
- ④ The start address of the single block is now transferred from the address start register to its permanent location in program register D.

Except for the RCW, all starting point addresses used in the block transfer operation are now in their permanent places. Note that both the RCW address and the start point of single block address (Figure 5.5-3) can be increased by one by use of the 1-upper circuit.

The following are Tran Control 2 functions (Figure 5.5-4):

- ⑥ Signal a memory access operation in which the RCW in the address specified by the address control A register is read out from core storage via the information bus to the auxiliary register.
- ⑦ Compare the start address of the RCW in Auxr 2 with the stop address of the RCW in Auxr 3. A complement add Auxr 2 to Auxr 3 adder operation is accomplished. The adder output is then tested for No-Carry (start greater than stop) and zero (start equal to stop). If the start address is greater than the stop address, an error is indicated. If the start address equals the stop address, the "block" to be gathered consists of just one word so a Match signal is generated. If the start address is less than the stop address, the operation proceeds. The contents of Auxr 2 and Auxr 3 are unchanged by the operation.
- ⑧ Form the address of the next RCW by serially shifting the digits of the RCW address in the address control register through the 1-upper circuit.
- ⑪ Transfer the address of the next word to be gathered from Auxr 2 to the address start register.

At this point, all "housekeeping" functions have been performed, and the actual gather operation can begin whereby the block of data specified by the RCW in the auxiliary register is transferred to the single block of locations, the first address being specified by the address in program register D.

In general, transfer control 3 causes the read out from core storage and the storage of each word is specified by the RCW. A test is made at the completion of each store operation to determine whether the stop address has been used. If it has been used, transfer control 2 functions are repeated and a new RCW is read into the auxiliary register. At the same time that the stop vs start address test is being made, the Auxr 2 address is increased by one. If start is less than stop, another word is read out from core storage and stored in the next sequential location in the single block. Now refer to Figure 5.5-5 for Transfer Control 3 operation.

The following are Tran Control 3 functions (Figure 5.5-5):

- ⑬ The contents of the location specified by the address start register are read out from core storage, via the information bus, to the arithmetic register by means of a normal memory access operation.

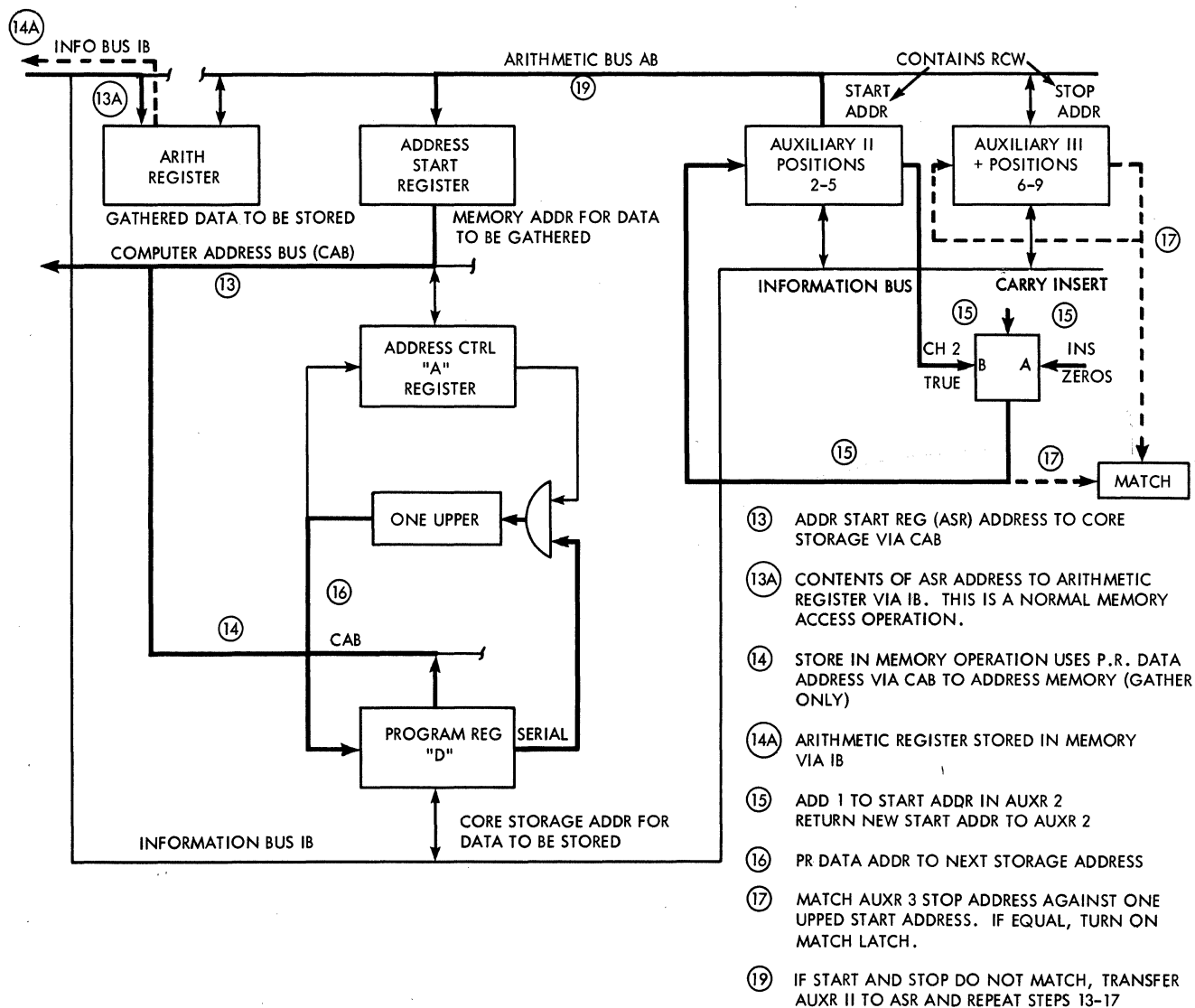


FIGURE 5.5-5. DATA FLOW FOR TRANSFER CONTROL 3 OPERATIONS--GATHER ONLY

- 14 A Store in Memory operation causes the arithmetic register contents to be stored in the location specified by the address in the PR D Register.
- 15 One is added to the start address in Auxr 2 to create the address for the next core storage readout operation.
- 16 The address in program register D is serially shifted right to the 1-upper circuit and the one upped address is returned to the PR D, ready to address memory on the next storage operation.
- 17 The Auxr 2, One Upped, Adder output of 15 is compared to the serial output of Auxr 3 by the Match circuitry. When a match is achieved, the Match latch is turned on. After a short delay, a Match Delay latch is turned on. The last word is transferred from core storage to the arithmetic register to the new core storage location due to the delay in turning on Match Delay. One of the following sequences then occurs, depending upon conditions:

- 19 a. If the RCW is +, the functions of transfer control 2 are repeated. The next RCW is read out from core storage and Transfer Control 3 functions are repeated using the new RCW.
- b. If the RCW is -, an end-data operation is signalled.

Circuits for Record Gather

Figure 5.5-2 shows a block diagram of the circuits which are summarized in positive logic form by Figures 5.5-6 a and b. The objectives and/or purposes of the various operations are shown in Figures 5.5-1, 2. The circuits to accomplish any specific objective can be followed by locating the corresponding circled number on Figure 5.5-6. A data flow is shown on Figure 5.5-2. The labels on the registers indicate the register usage during transfer control 3 time. Figure 5.5-7 shows the sequential timing for the operation.

Introduction and Circuits for Record Scatter

As previously noted, circuits for recordgather and record scatter are almost identical. Figure 5.5-8 is a block diagram of the scatter operation. The symbols * and ** on Figure 5.5-6 b indicate the circuits which are different. When data is to be scattered from the single block, core storage is addressed by the address in program register D at D Ctrl 1 time (Instead of ASR). Core storage is addressed, for scattering of data from the single block in the storage operation, by the contents of the ASR (Instead of the PR D). Figure 5.5-3, 4 and 9 show the data flow during transfer control 1, 2 and 3 for the record-scatter operation.

5.5.02 Edit Numerical to Alphameric

This is a record-scatter operation which operates in much the same manner as Op + 65 described in the previous section. Review section 5.5.01 before proceeding with the following discussion.

Op +56 Edit Numerical to Alphameric (ENA)

Blocks of numerical data are converted from the normal single-digit representation to the double digit alphameric representation. The sign of the numeric word is ignored in this operation. For example, the numeric word

- 00145		62001
/		/
in the single block becomes		
↓		↓
9090919495	and	9692909091

in two successive core storage locations in the scatter area defined by the record definition word. Note that in setting up the record definition words, two core storage locations are required for each converted numeric word.

Note: Circled Numbers apply to text and Figures 5.5-1, 2, 3, 4 and 5
Assume Start address is one less than Stop address and RCW sign is +

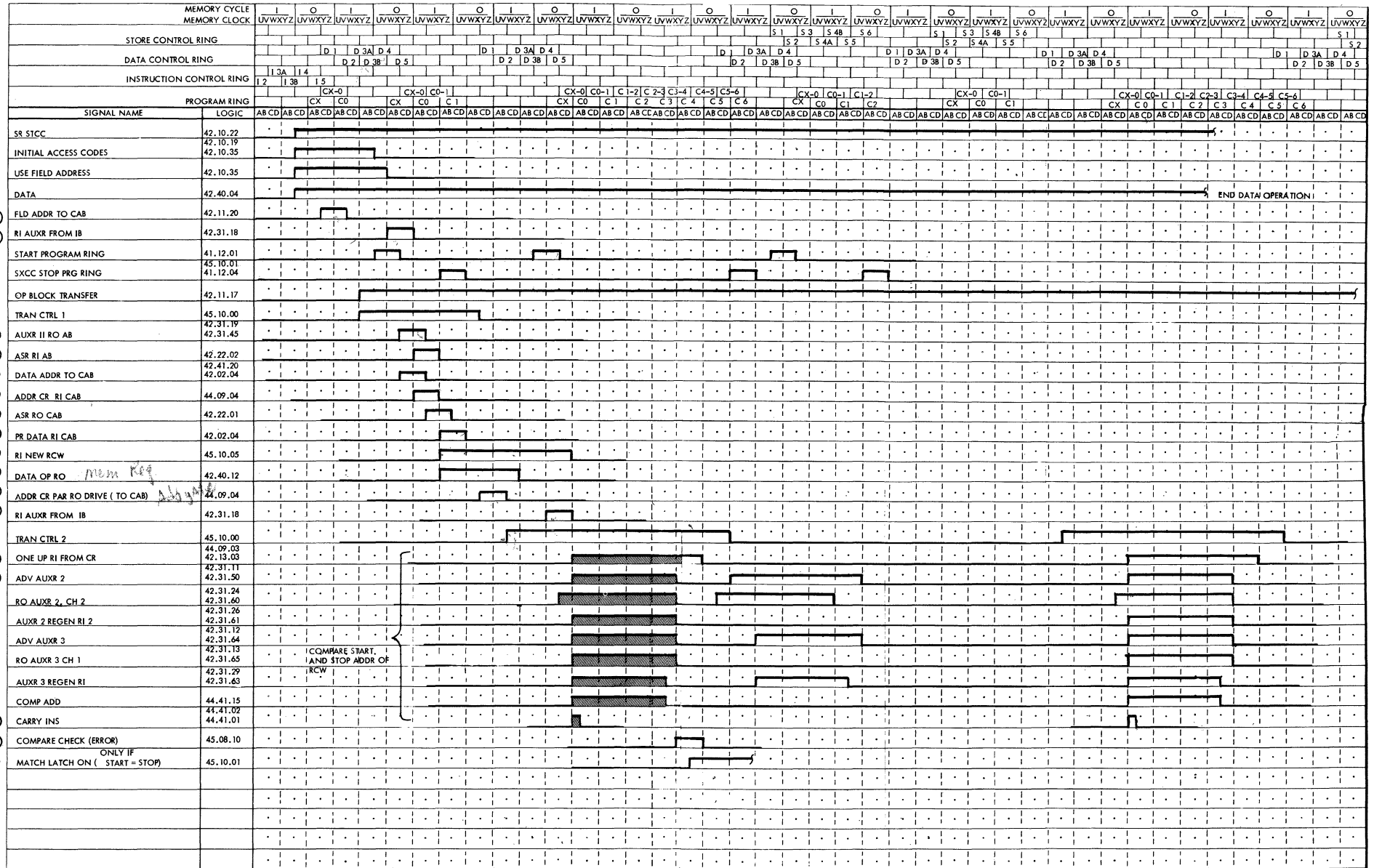


FIGURE 5.5-7A. SEQUENCE FOR BLOCK TRANSFER

Note: Circled Numbers apply to text and Figures 5.5-1, 2, 3, 4 and 5
Assume Start address is one less than Stop address and RCW sign is +

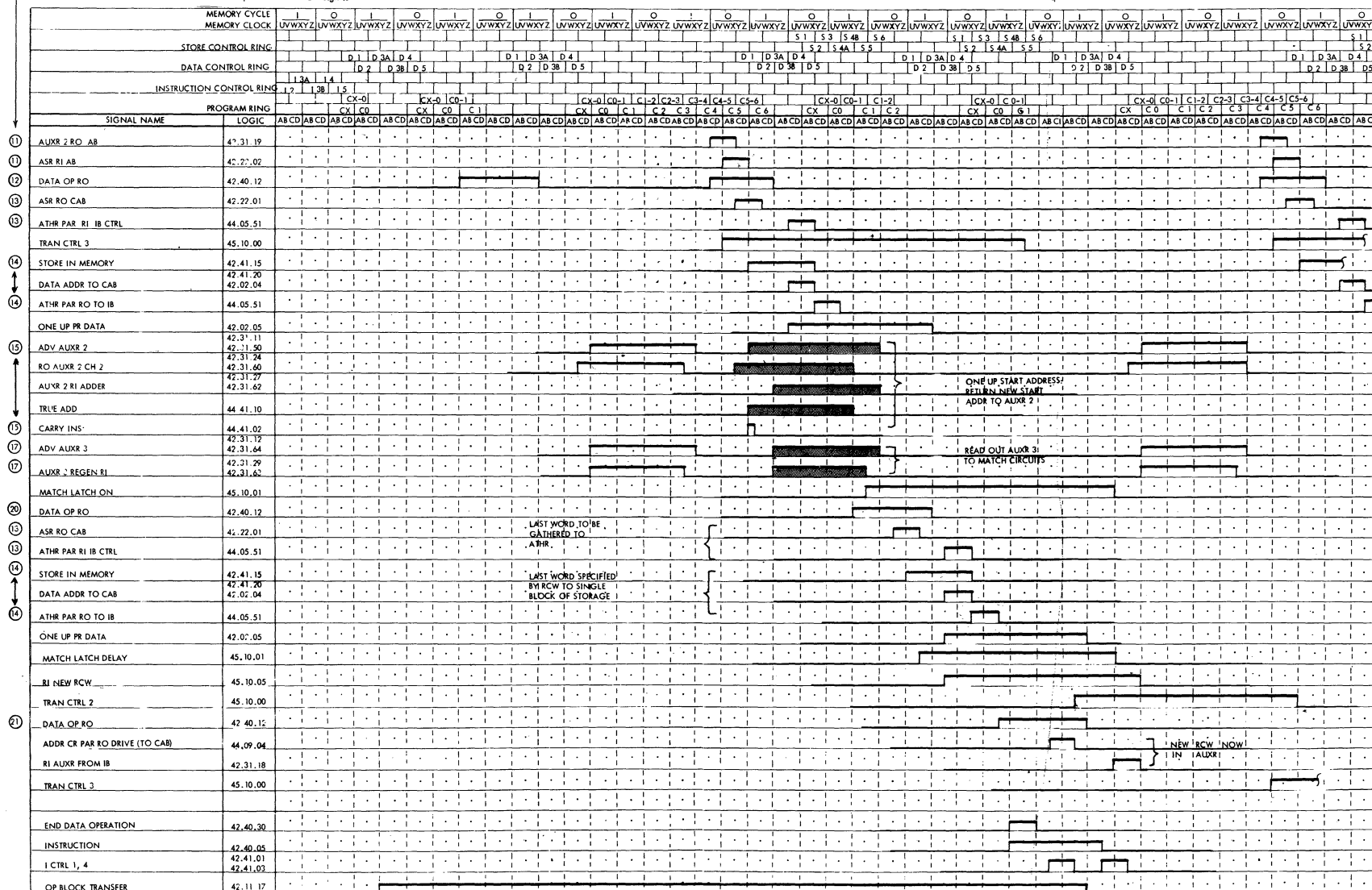


FIGURE 5.5-7B. SEQUENCE FOR BLOCK TRANSFER

data cycle

data cycle

data cycle

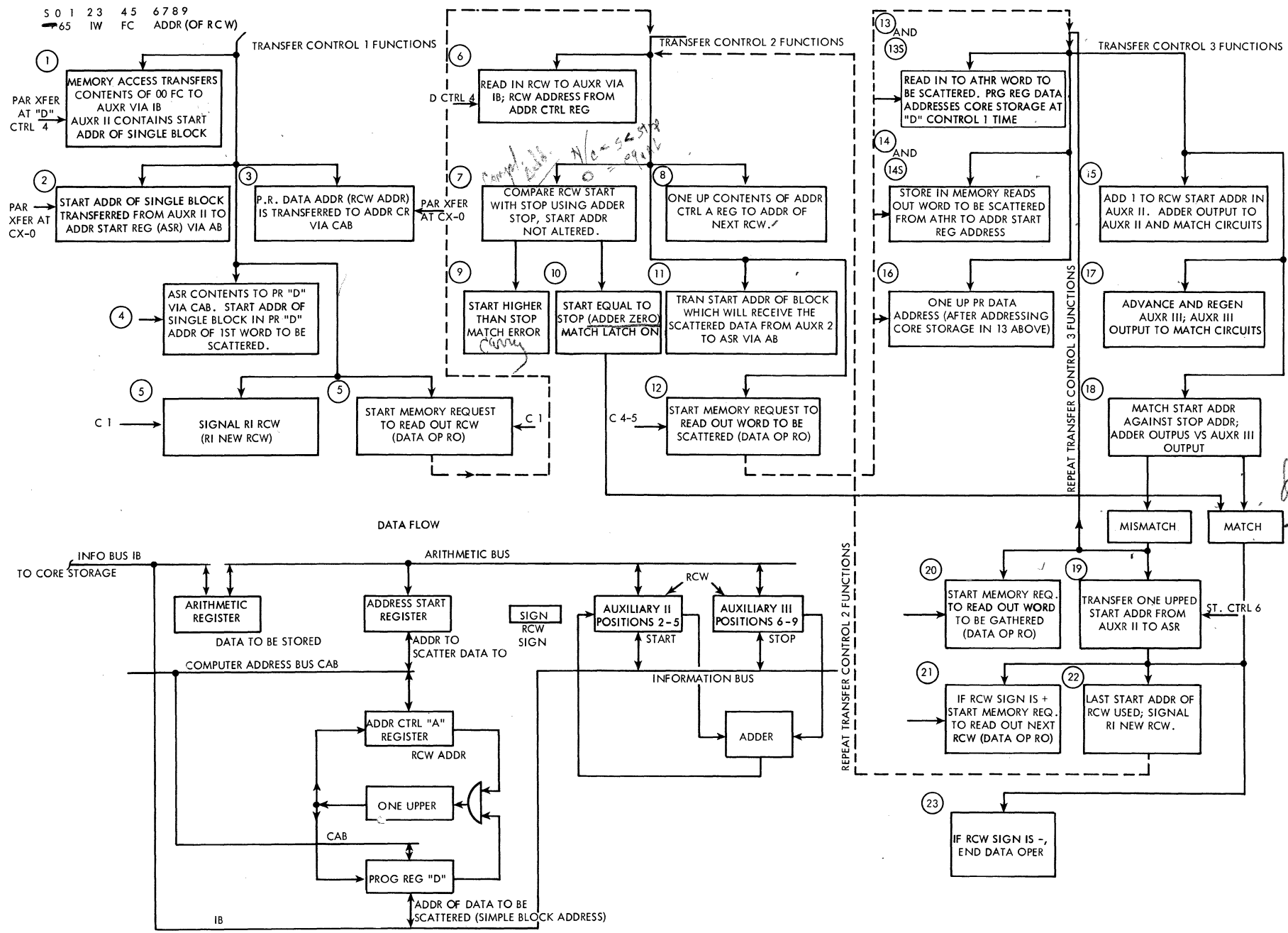


FIGURE 5.5-8. BLOCK TRANSMISSION--RECORD GATHER

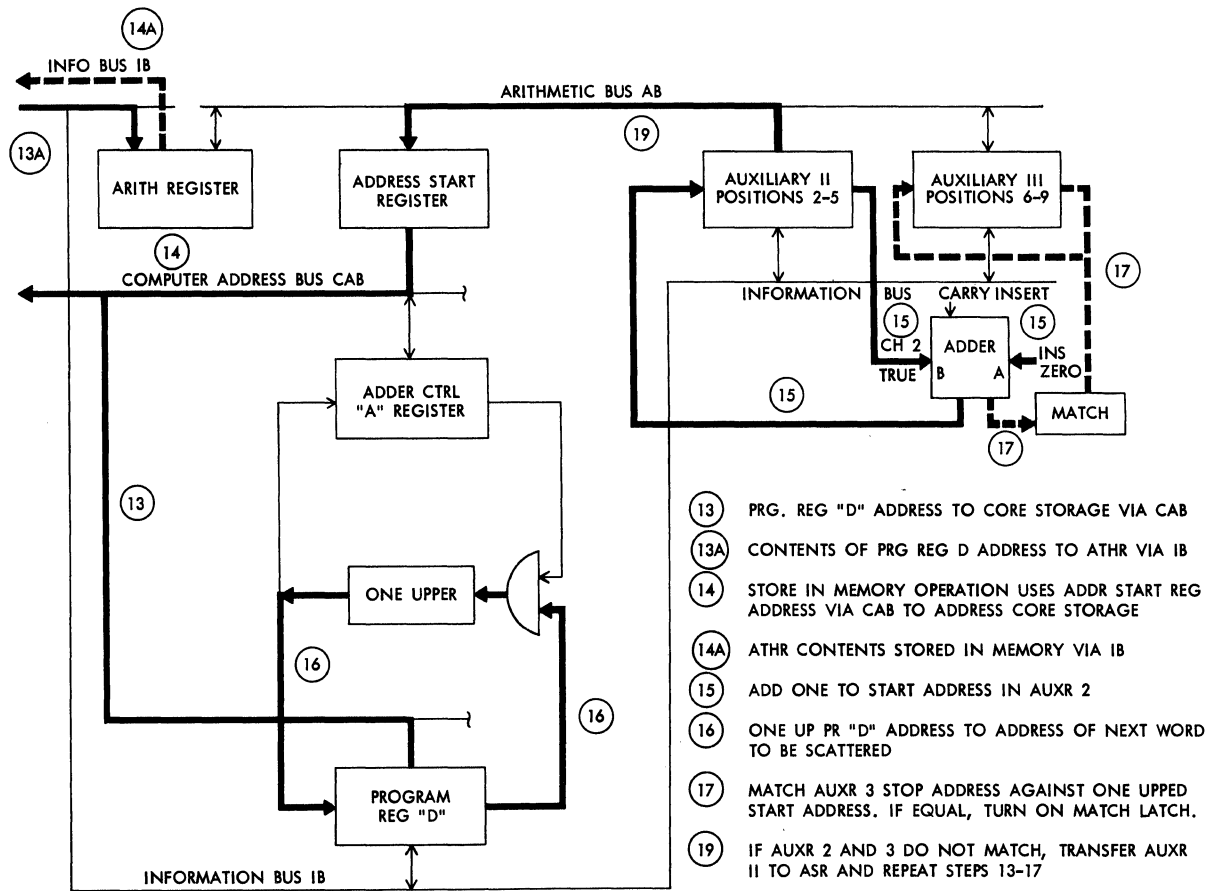


FIGURE 5.5-9. DATA FLOW FOR TRANSFER CONTROL 3 OPERATIONS—SCATTER ONLY

Introduction to Edit Numerical to Alphameric

The Edit Num to Alpha operation is described with the aid of Figure 5.5-10 which is a flow chart of the operation. The circled numbers on Figure 5.5-10 refer to text, logic on Figure 5.5-11, and the sequence chart on Figure 5.5-12 a, b, and c.

The general sequence of operation, as illustrated by Figure 5.5-10, is as follows:

- ①-④ These are normal Transfer Control 1 functions which accomplish the following:
 - a. Place the address of the first RCW in the Address Control Register.
 - b. Place the starting address of the single block, from which data is to be scattered, in program register D.
- ⑤-⑩ These are Transfer Control 2 functions which accomplish:
 - a. Transfer of the RCW from core storage to the Auxr.
 - b. Test the RCW for the start address less than the stop address. If start is less than stop, the start address for the scattered block is transferred to the ASR.
 - c. The address control register is 1-upped to create the address of the next RCW.
 - d. An error is indicated if start is equal to or greater than stop.
- ⑪-⑰ Control conversion of digit positions 0 - 4 of the numeric word to the first alphameric word.
- ⑱-⑳ Control conversion of digit positions 5 - 9 of the numeric word to the second alphameric word.
- ⑪, ⑫ The numeric word to be converted is transferred to the arithmetic register from the core storage location specified by program register D. Digit positions 4, 3, 2, 1, 0 of the Athr are then scanned out to the WBR at alternate program ring times. For example, 4th position bits are read out at C1 time, 3rd position bits at C3 time, 2nd position bits at C5 time, and so forth. Nines are inserted at C2, C4, C6, C8 and C10 time. The bits are shifted right in the word buffer register.
- ⑬, ⑭ The newly created alphameric word is now transferred back to the arithmetic register, replacing the numeric word. The alphameric word is stored in the location specified by the address start register.
- ⑮, ⑰ Because the alphameric word to be created from digits 5 - 9 of the numeric word must be stored in the next sequential core storage location, the start portion of the RCW in the Auxr must be increased by one. This is accomplished by a true-add operation in which the "one" is added by forcing a carry insert. At the completion of the one up operation, the new start address is transferred from Auxr 2 to the address start register in preparation for the conversion of digits 5 - 9.

- ①⑥ Whenever the start address is increased, it is matched against the stop address to determine when the last location is used. The Match latch is turned on but is not tested until the remainder of the numeric word digits (5 - 9) are converted during Tran Ctrl 5 time (Figure 5.5-10).
- ①⑧, ①⑨ The numeric word to be converted is transferred back to the Athr from the location specified by program register D. Digit positions 9, 8, 7, 6, 5 of the Athr are scanned out to the WBR at odd program ring times; position 9 bits at C1 time, position 8 bits at C3 time, and so forth. Nines are inserted, as before, at even program ring times; C2, C4, C6, C8, C10.
- ②②, ②① The alphameric word is transferred back to the arithmetic register and is stored in the core storage location specified by the ASR (1- upped since steps ①③ and ①④).
- ②②, ②⑥ The start address in Auxr 2 must be increased by one in preparation for the next numeric word conversion. At the end of Tran Ctrl 5 time, the new start address is transferred to the address start register in preparation for the next numeric word to be converted.
- ②③ The address in program register D is upped by one in preparation for transferring the next numeric word from core storage to the arithmetic register.
- ②④, ②⑤ While the 1-up operation of ②② is taking place, the start address is matched against the stop address. The two will never match at this point, provided an even number of storage locations are specified by the RCW in the auxiliary register. An error is indicated if the two do match. If the Match latch was not turned on in ①⑥, the functions of transfer control 4 and 5 are repeated.
- ②⑦ If the sign of the current RCW is + and a start stop match is attained, a new RCW is transferred to the auxiliary register. The usual tests and housekeeping operations of transfer control 2 are performed and the Num-Alpha conversion process continues under control of the new RCW.
- ②⑧ If the sign of the current RCW is minus when a match is attained, an end-data operation is signalled.

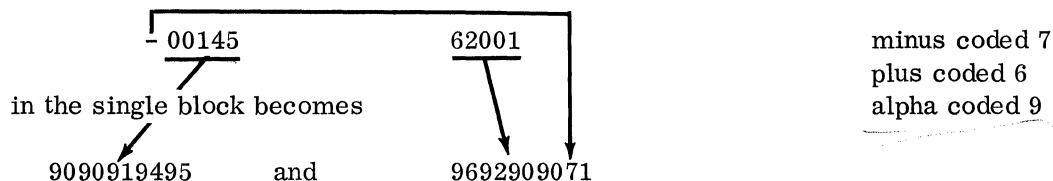
Circuits for Edit Numerical to Alphameric

Figure 5.5-11 shows the logic used to accomplish the objectives shown on Figure 5.5-10 and discussed in the preceding section. The circuits to accomplish any specific objective can be followed by locating the corresponding circled number on Figure 5.5-11. Figure 5.5-12 shows the sequential timing for the operation. Figure 5.5-13 outlines the data flow path for serially shifting the arithmetic register contents into the word buffer register.

5.5.03 Edit Numerical to Alphameric with Sign Control

Op -56 Edit Numerical to Alpha with Sign Control (ENS)

This code operates in the same manner as Edit Numerical to Alphameric except that the sign is examined and the proper digit instead as indicated. The numeric word



in two successive storage locations in the scatter area defined by the record definition word.

Introduction and Circuits

Circuits to accomplish sign control are illustrated in Figure 5.5-13. The sequence of operation is shown in Figure 5.5-15. The circled numbers in the following material refer to Figures 5.5-13, 15.

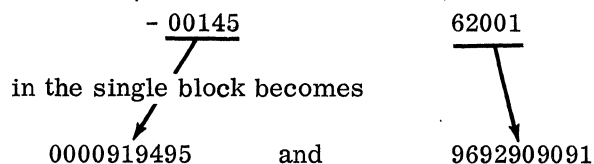
- ① The sign of the numeric word in the arithmetic register is transferred to the Op Sign latches via the arithmetic bus.
- ② Sign bits are inserted in the 8th position of the second Alpha word during Tran Ctrl 5 time.
- ③ The normal 9 insert in the 8th position of the second Alpha word is inhibited when the sign of the Athr is + or - so that 6 or 7 can be inserted.

5.5.04 Edit Numerical to Alphameric with Blank Insertion

This is a record-scatter operation which functions in much the same manner as Op +65, described in section 5.5.01. The blank insertion feature operates as described below.

Op +57 Edit Numerical to Alphameric with Blank Insertion (ENB)

In addition to the normal functions of Edit Numerical to Alphameric, initial zeros in the numeric word are converted to blanks (00) in the alphameric word. The numeric word



in two successive storage locations in the scatter area.

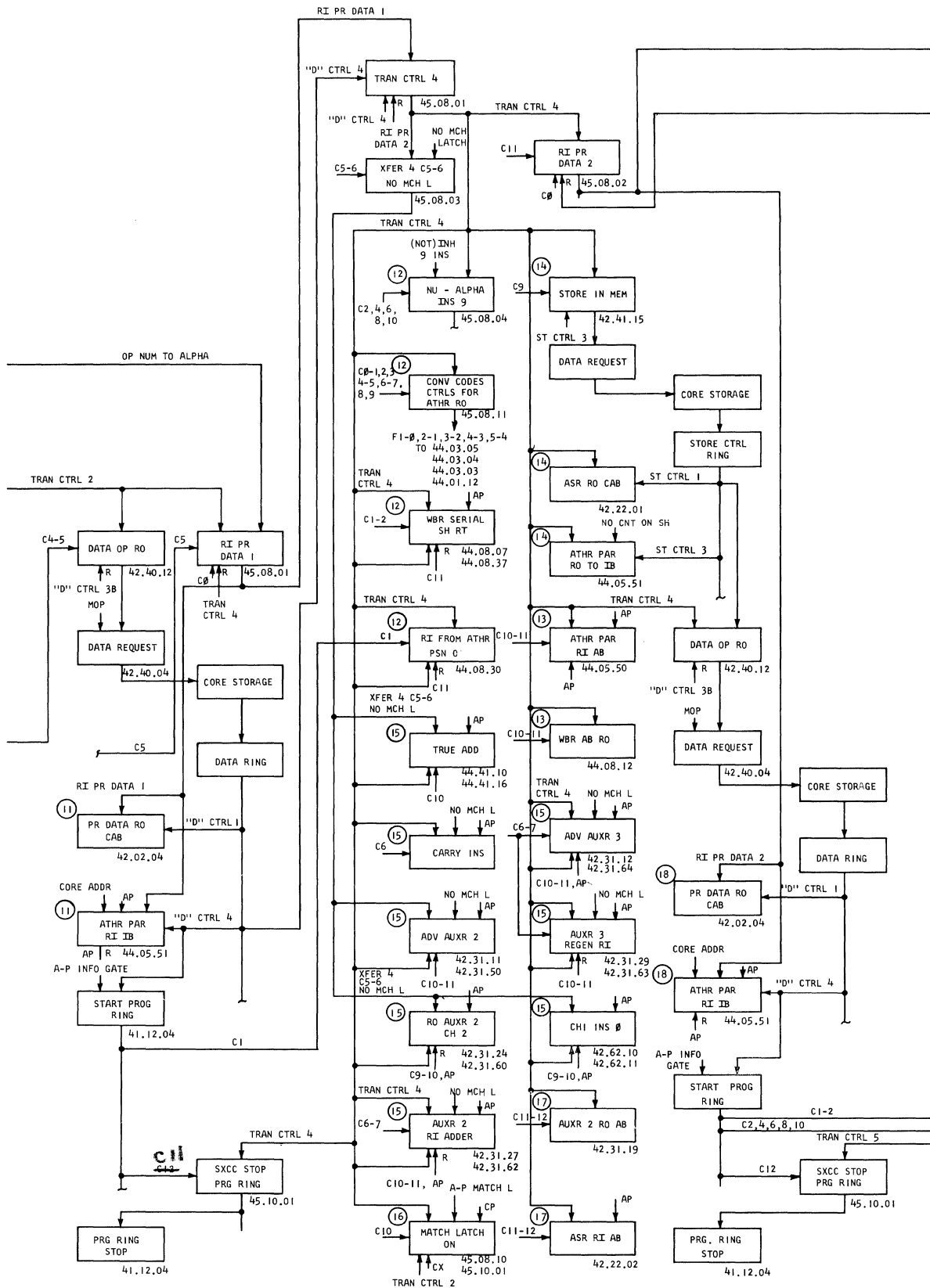


FIGURE 5.5-11B. EDIT NUMERICAL TO ALPHAMERIC LOGIC

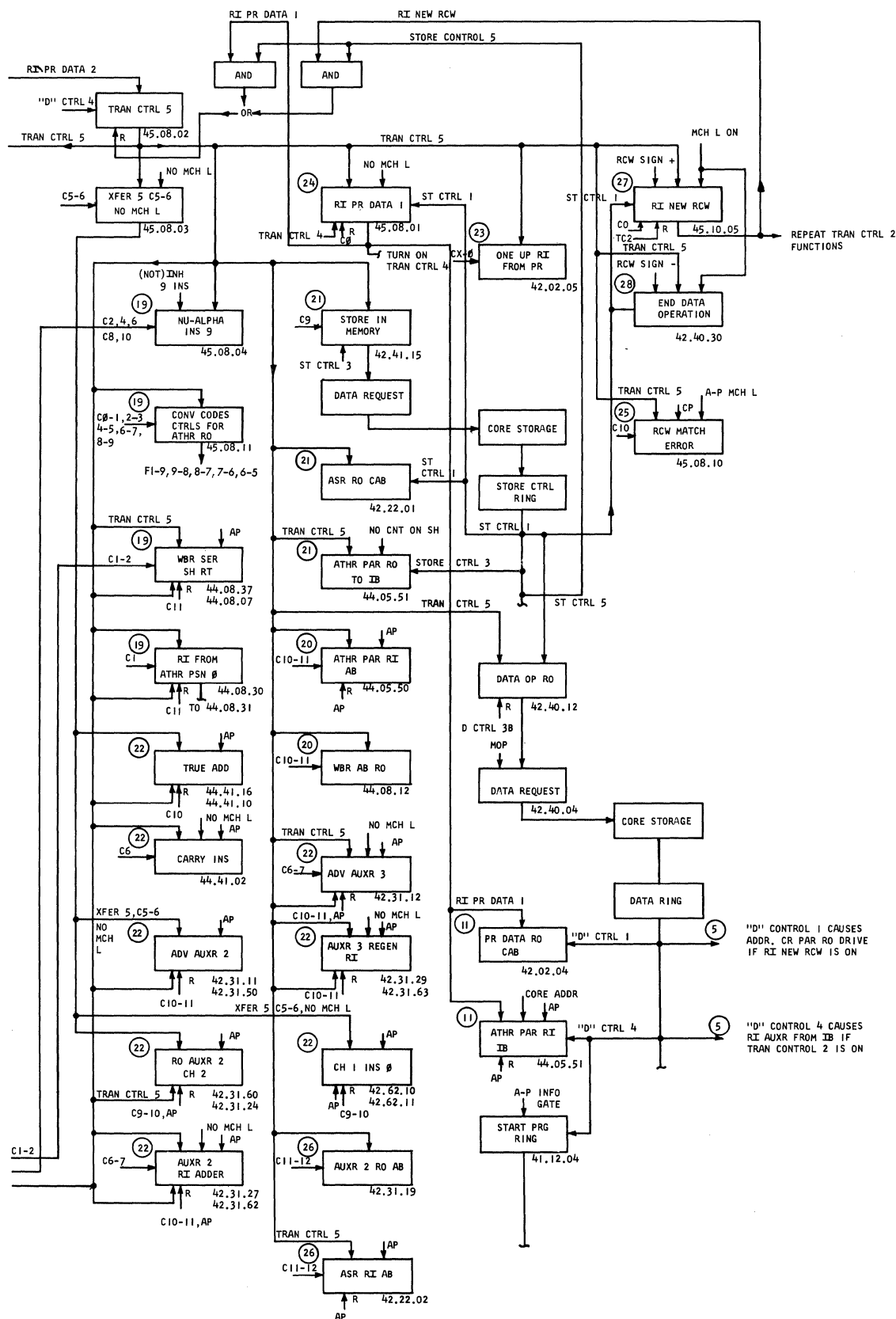
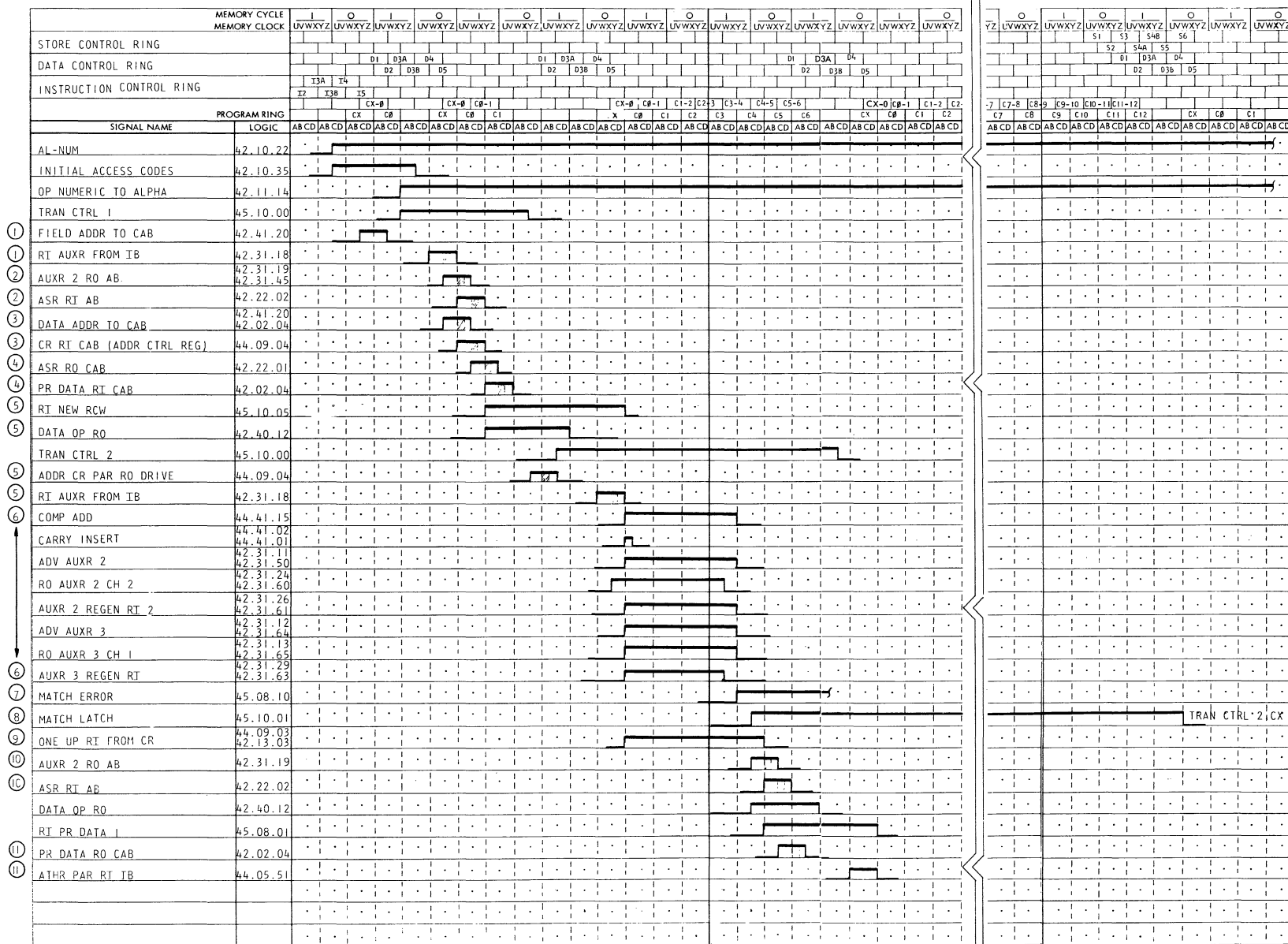


FIGURE 5.5-11C. EDIT NUMERICAL TO ALPHAMERIC LOGIC

FIGURE 5.5-12A. SEQUENCE FOR EDIT NUMERICAL TO ALPHAMERIC



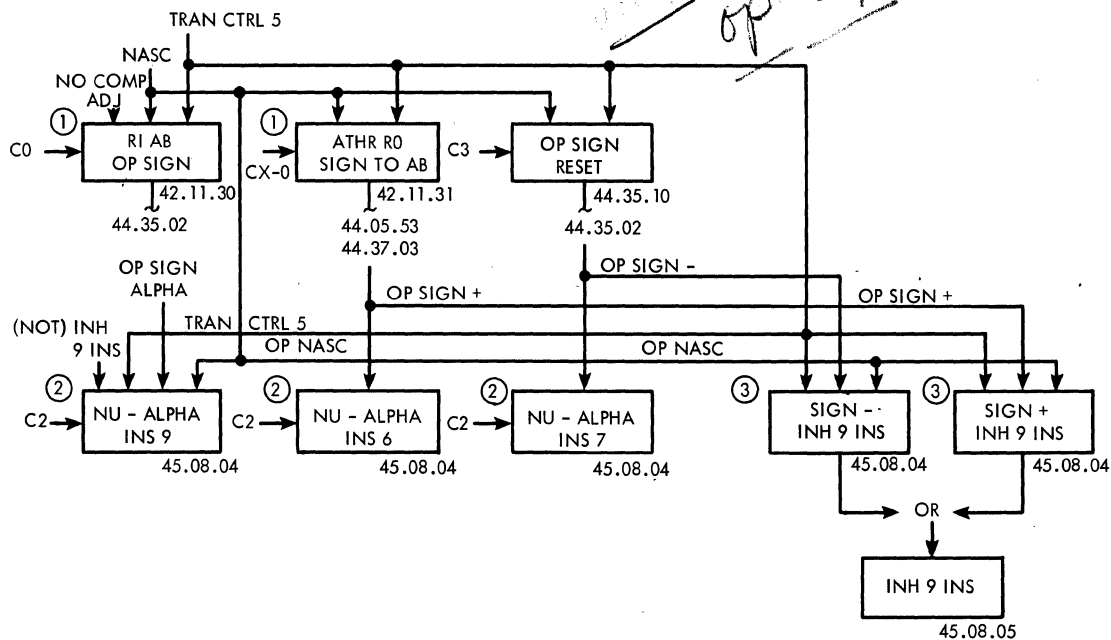


FIGURE 5.5-13. LOGIC FOR EDIT NUMERICAL TO ALPHAMERIC WITH SIGN CONTROL--OP NASC

Introduction and Circuits

Circuits to accomplish blank insertion are illustrated in Figure 5.5-14. The sequence of operation is shown in Figure 5.5-15. The circled numbers in the following text refer to Figures 5.5-14 and 15.

The following steps are performed in inserting blanks (zeros) to the left of the first significant figure.

- ① Transfer the numeric word to be scattered from the Athr to the Significant Digit Scanner via the arithmetic bus.
- ② Read Out the high order position of the SDS to the shift register.
- ③ Turn on the shift register output latches corresponding to the 9's complement of the highest order significant digit position. For example, if position 6 of the numeric word contains a significant digit and positions 0 - 5 are zero, the 0 and 3 bit Shift Register Output latches are turned on.

Zero are inserted in place of nines in the even digit positions of the word buffer register, under control of the SDS and shift register output, in the following cases.
- ④ SDS Register 0. When all positions of the numeric word are zero, the control latch for NABI remains on through both Tran Ctrl 4 and 5 time. This causes the inhibiting of 9's and the insertion of zeros at C2 through C10 time of both Tran Ctrl 4 and 5.

SDS Set high order significant digit
 "9's" Compl. of high order significant digit set in pos. O.P. latches of Shift Reg.

Insert "0's" instead of "9's" on Blanks in old pos.

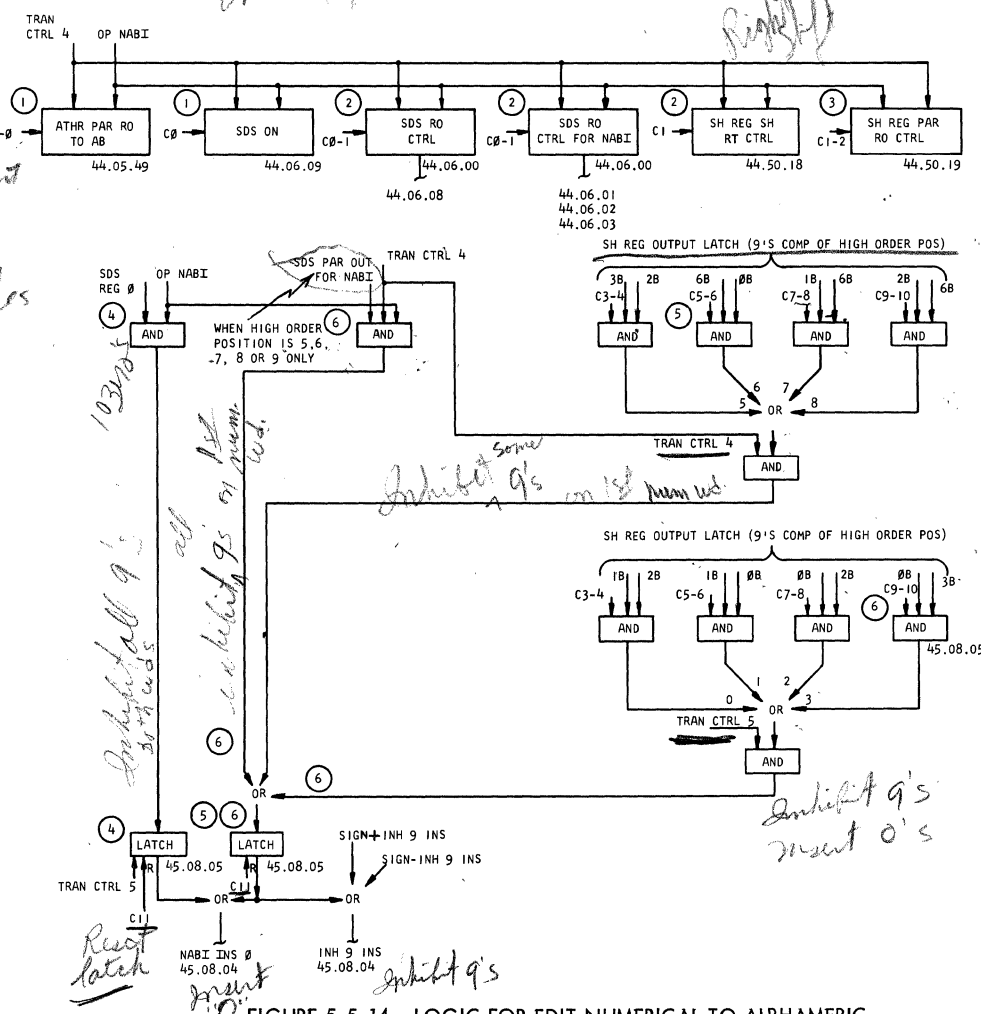


FIGURE 5.5-14. LOGIC FOR EDIT NUMERICAL TO ALPHAMERIC WITH BLANK INSERTION--OP NABI

- ⑤ SDS Register high-order position 0, 1, 2, 3, 4. For example, assume that the high order position is 3 (Figure 5.5-15). The 9's complement, or 6 and 0, Sh Reg Output latches are turned on. This causes the NABI Ins 0 control latch to come on at C5-6 time and remain on thru C10 of Tran Ctrl 4. Thus zeros are inserted at C6, C8 and C10 time in positions 0, 1 and 2. Nines are entered at all other even digit program ring times during Tran Ctrl 4 and 5.
- ⑥ SDS Register high-order position 5, 6, 7, 8 or 9. Whenever the SDS indicates the high-order position is 5, 6, 7, 8 or 9, zeros must be inserted in positions 0, 1, 2, 3 and 4 during Tran Ctrl 4 time, and in additional positions during Tran Ctrl 5 time. Insertion of zeros during Tran Ctrl 4 time is accomplished by OR-ing positions 5 - 9 of the SDS Par Out for NABI in order to turn on NABI Ins 0 for C2, 4, 6, 8 and 10 time of Tran Ctrl 4. Thus, positions 0 - 4 receive zero inserts. During Tran Ctrl 5 time zeros are inserted as shown on Figure 5.5-15, ⑥. For example, if the highest order significant digit is 6, the 0 and 3 bit Shift Register Output latches switch with C9-10 to allow the NABI Ins 0 control latch to be on only for the C10 time entry of a zero in position 5. Thus, positions 0 - 5 receive zero inserts and positions 6, 7, 8, 9 receive 9 inserts.

5099000000

5.5.05 Edit Alphameric to Numerical

This is a gathering operation which functions in a manner similar to Op -65. Record Gather, described in Section 5.5.01.

Op ⁻⁵⁷~~-56~~ Edit Alphameric to Numerical (EAN)

Alphameric words are gathered from scattered locations in core storage, under control of record definition words. Two consecutive alphameric words become a single numeric word. Alpha or plus signs are translated as +, minus signs are translated as -, and blanks are translated as zero's. For example,

	0000949792	and	9197909063
become +	00472		17003

Introduction and Circuits

①, ⑩ Figure 5.5-16 outlines the edit alpha to numeric process. Steps 1 through 10 are the normal record scatter-gather transfer control 1 and 2 functions.

⑪ The first alpha word to be translated to numeric is brought to the arithmetic register by a memory access operation.

⑭ The Athr bits in digit positions 1, 3, 5, 7, and 9 (in that order) are serially scanned out of the arithmetic register to the 9th position of the word buffer register. The bits are then shifted left in the word buffer register. Using the numbers in the previous example, the register would appear as follows at the end of step ⑭

Arithmetic Register	0000949792
Word Buffer Register	0000000472

⑫, ⑬ While the odd digits of the first Alpha word are being scanned out of the Athr, the start address in Auxr 2 is one upped and is tested for match with the Stop address. The new start address is transferred to the ASR. The second of the alpha words to be translated is then read out of the ASR address to the Arithmetic Register.

⑰ The Athr bits (from the second alpha word) in digit positions 1, 3, 5, 7, 9 are serially scanned out of the arithmetic register to the 9th position of the word buffer register. As the bits from the second alpha word enter at position 9 of the WBR, the entire register contents are left shifted. The registers appear as follows at the end of stage ⑰

Arithmetic Register	9197909063
Word Buffer Register	0047217003

⑳ Position 8 of the second alpha word is scanned out of the arithmetic register at C 4-5 time as shown on Figure 5.5-18. The Athr output latches 0, 1, and 3 bits define the sign as follows:

- ②①, ②③ The completed numeric word is now transferred to the Arithmetic Register and is then stored in the address specified by program register D.
- ①⑦, ②②
②④ The newly 1-upped start address is transferred to the address start register in preparation for the gathering of the next alpha word. The address in program register D is 1-upped to create the address for the storage of the next numeric word.

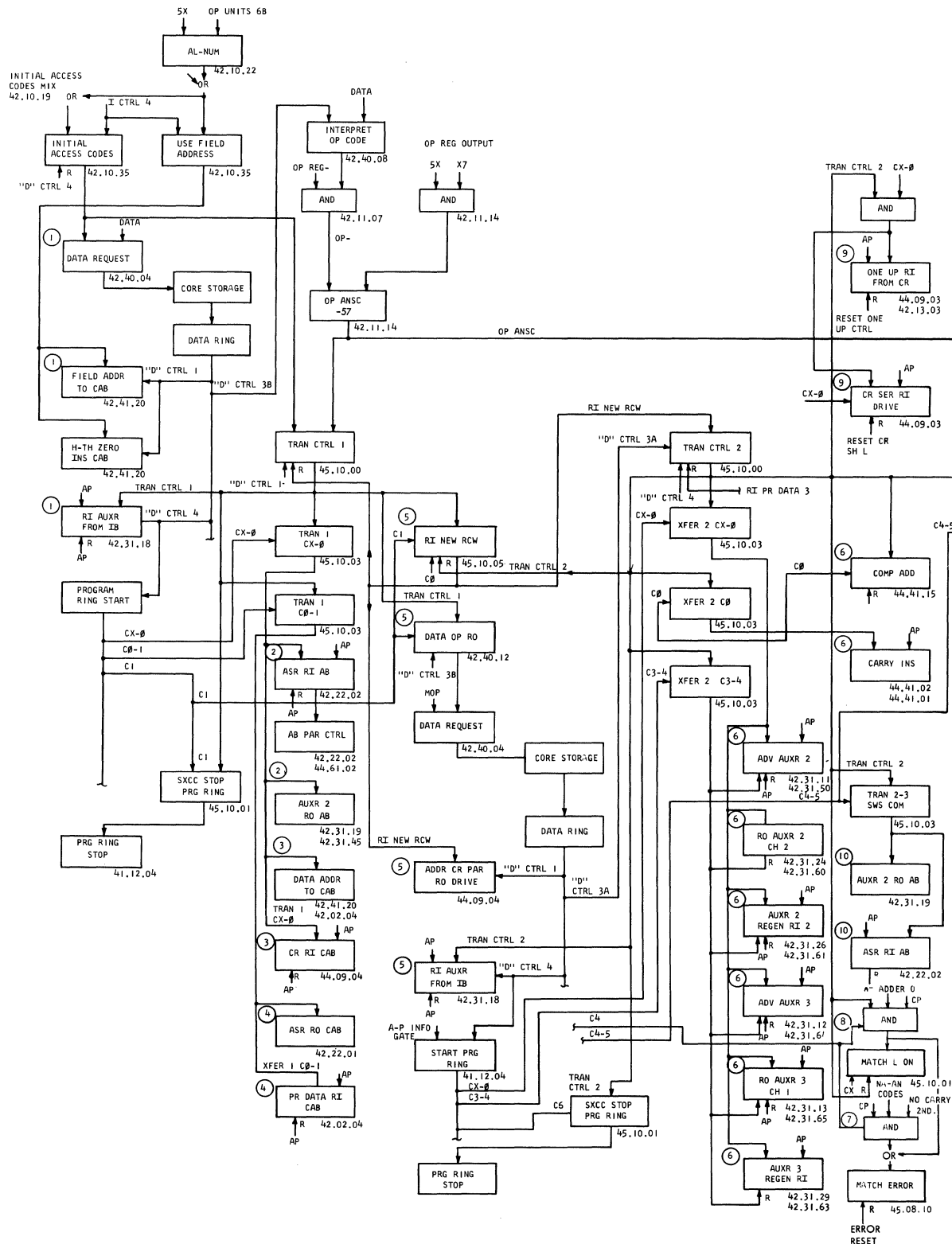


FIGURE 5.5-17A. LOGIC FOR EDIT ALPHAMERIC TO NUMERICAL

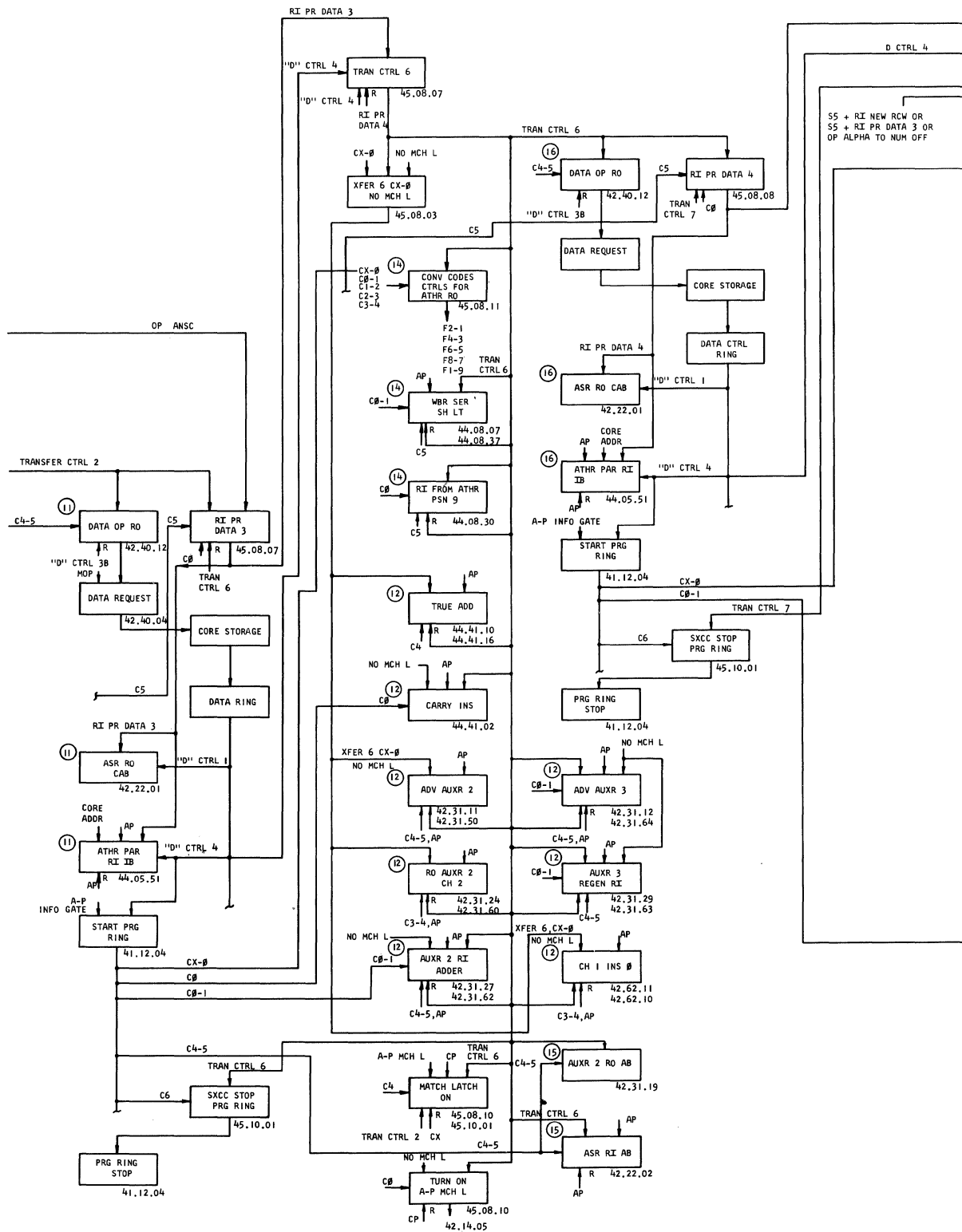


FIGURE 5.5-17B. LOGIC FOR EDIT ALPHAMERIC TO NUMERICAL

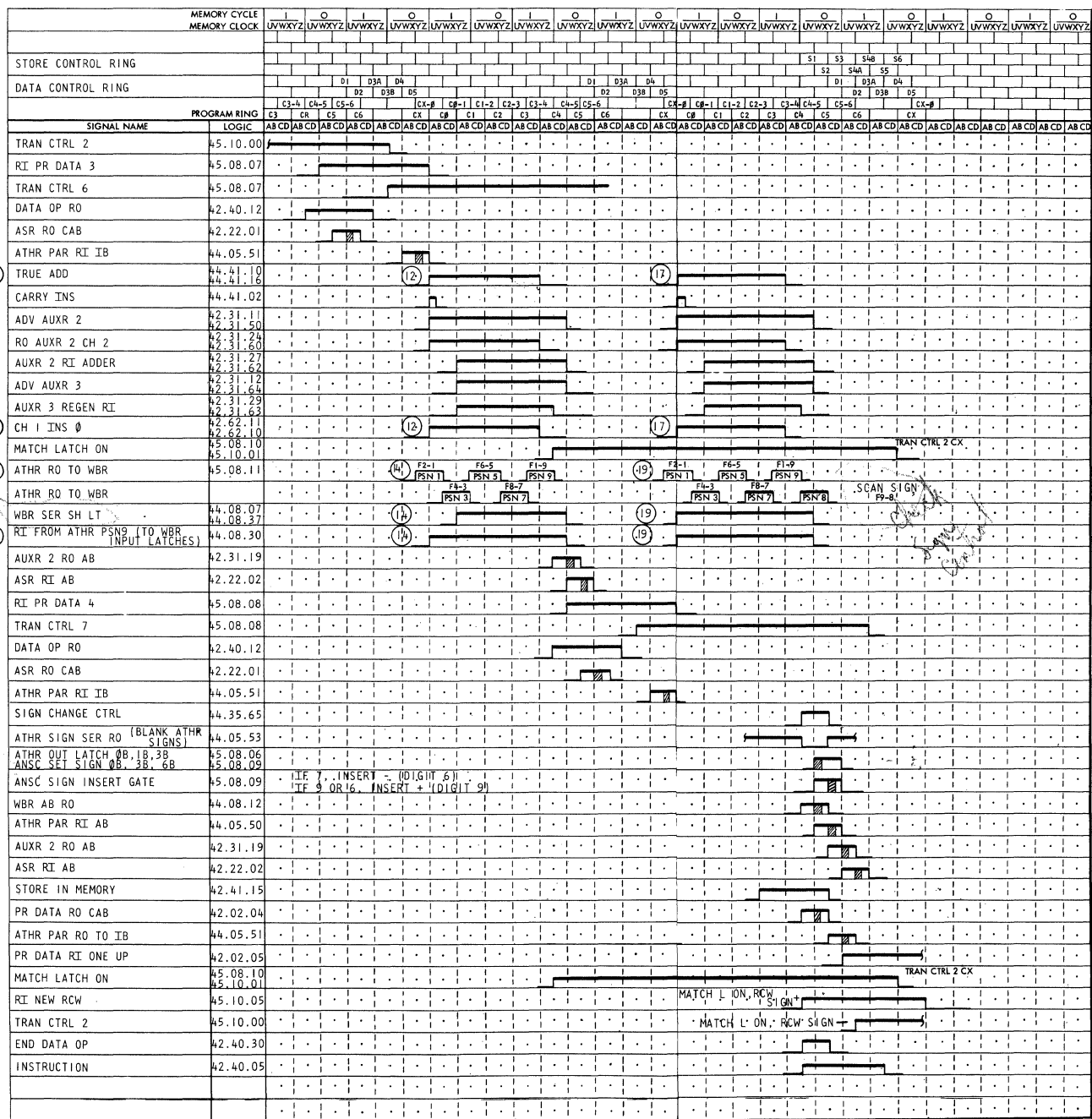


FIGURE 5.5-18. SEQUENCE FOR EDIT ALPHAMERIC TO NUMERICAL

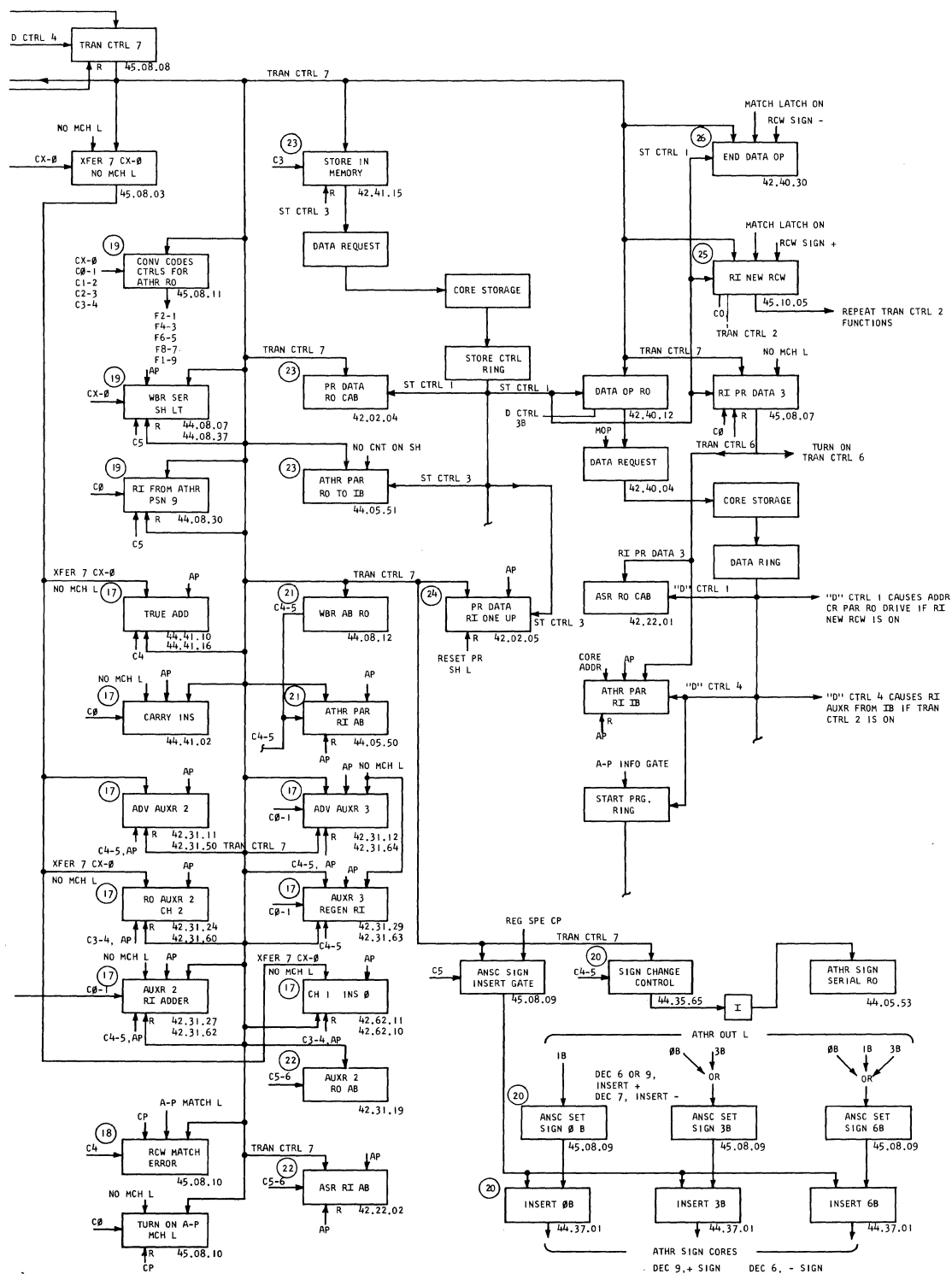


FIGURE 5.5-17C. LOGIC FOR EDIT ALPHAMERIC TO NUMERICAL

5.6.00 COMPARE INSTRUCTIONS

Each of the compare codes automatically turns off the high, equal, and low indicators, and then turns one of them on again, depending upon the result of the comparison.

5.6.01 Compare Accumulator to Storage

Op + 15, + 25, + 35 (C#) Compare Accumulator to Storage

The entire contents of the accumulator are compared with the word in the core storage location specified by positions 6 - 9 of the program register. Only the portion of the storage word designated by field definition is compared. Signs are included in this comparison. The following chart shows the relative values of signs and digits in compare operations.

Highest	+ 9999999999
	to
	+ 0000000000
	- 0000000000
	to
	- 9999999999
\	
Lowest	Alpha 9999999999
	to
	Alpha 0000000000

The high, low, or equal indicators (i. e. , latches) are set by whether this accumulator is high, low, or equal in relation to the storage word; e. g. , if the accumulator is lower, the low indicator is set. The accumulator and the storage word are unchanged by the operation.

Introduction (Figure 5.6-1)

Compare accumulator is an initial access code, requiring the read-out from core storage of the word specified by positions 6 - 9 of the program register. A normal data request operation causes the selected core storage word to read out to the information bus. The arithmetic register is signalled to read in from the I. B. Thus, the factor to be compared against the accumulator is brought to the arithmetic unit.

The interpret op code signal is brought up in the usual manner during the data request operation. This, in turn, controls both arithmetic codes and compare codes signals which set up the actual compare operation. The arithmetic code signal initiates the transfer of the designated accumulator word (i. e. , code + 15 designates accumulator 1) to the auxiliary register via the AB at the same time as the core storage word is being transferred to the arithmetic register via the IB. Figure 5.6-1 shows a schematic block diagram of the operation.

The comparison of the arithmetic register (core storage) against the auxiliary register (accumulator) is discussed in two parts as follows:

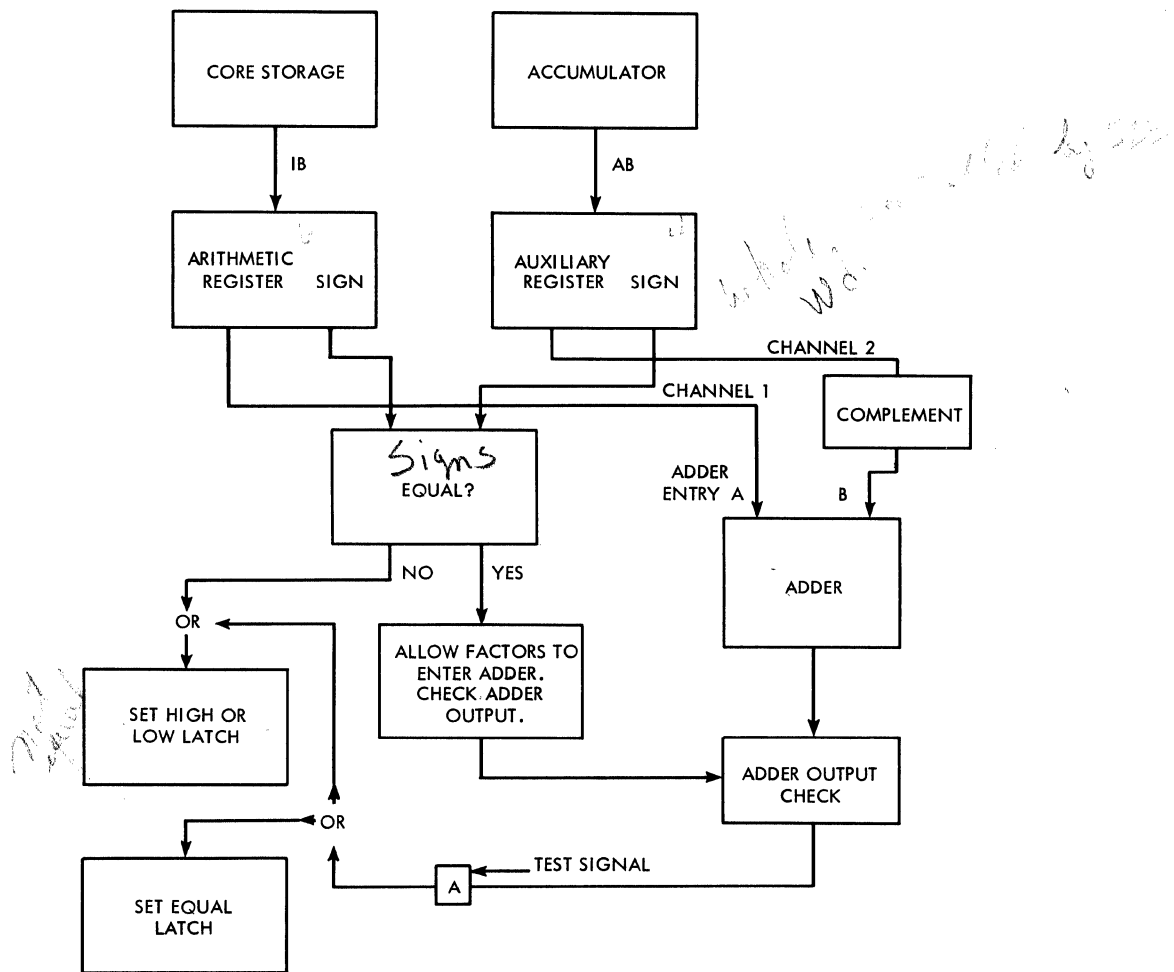


FIGURE 5.6-1. COMPARE ACCUMULATOR BLOCK DIAGRAM

Seq. Chart 31.00.64

Signs Unequal. The signs of both arithmetic register and auxiliary register may be plus or minus or alpha. The sign condition of the two registers is compared. If they are equal (-and-, + and +, alpha and alpha), the two factors must be complement added to determine which is the lowest. If the signs are unequal, the high or low indicator latch is turned on, and the operation ends without performing the adder operation.

Signs Equal. When the signs are equal minus, equal plus or equal alpha, further testing is necessary to determine whether the accumulator is high, low, or equal. The number in the arithmetic register is added to the complemented number from the auxiliary register. The signs of the two numbers are not considered. The following results are possible:

1. Adder Output Carry. If the auxiliary register value (accumulator) is equal to or lower than the arithmetic register.
2. Adder Output No Carry. If the auxiliary register value (accumulator) is higher than the arithmetic register.

3. Adder Zero. If no significant digits appear at the adder output, the adder zero latch remains on at test time.
4. Adder Non-Zero. The first significant digit at the adder output turns off the adder zero latch and turns on the non-zero latch.

The accumulator field size relative to the storage word field size can be determined with the following results:

5. Arithmetic Register smaller than Accumulator (Auxr-Reg). If the field ring is driven from its starting position to its end before the shift register has signalled that the last digit has left the auxiliary register, the arithmetic register has fewer digits than the auxiliary register.
6. Accumulator Less than or Equal to Arithmetic Register. If the field ring reaches its end position at the same time as the shift register signals that the last accumulator digit has left the auxiliary register, the two factors are of equal field length. Similarly, if the shift register signals end of accumulator digits before the field ring ends, the accumulator has fewer digits than the arithmetic register.

In addition to these 6 factors, the factor sign signals, equal-, and equal + or alpha, are considered in determining whether the high, equal or low indicator latch should be turned on. This switching is necessary because, for example, - 999 is less than -998, while + or alpha 999 is greater than + or alpha 998. Recall that signs are not considered during the complement addition comparison operation.

The test to turn on the high, low or equal latches comes when the results of the high-order field digit (arithmetic register digit) comes out of the adder. The test signal is initiated by field stop match when the field ring advances until it agrees with the tens position digit in the field register.

Circuits for Compare Accumulator to Storage

A block diagram showing the logical circuitry required for the compare operation is shown in Figure 5.6-2 and sequence charts in Figures 5.6-4 and 5.6-5. The following major objectives must be accomplished:

Read Out the Data Word From Memory

	<u>Signal</u>	<u>Time</u>
Gates Required:	Data Address to CAB	"D" ctrl. 1
	Athr. Par. RI IB	"D" ctrl. 4

Transfer Accumulator to Auxiliary Register

Gates Required:	A1 RO AB	"D" ctrl. 4
	Auxr. 1, 2, 3, RI AB	"D" ctrl. 4

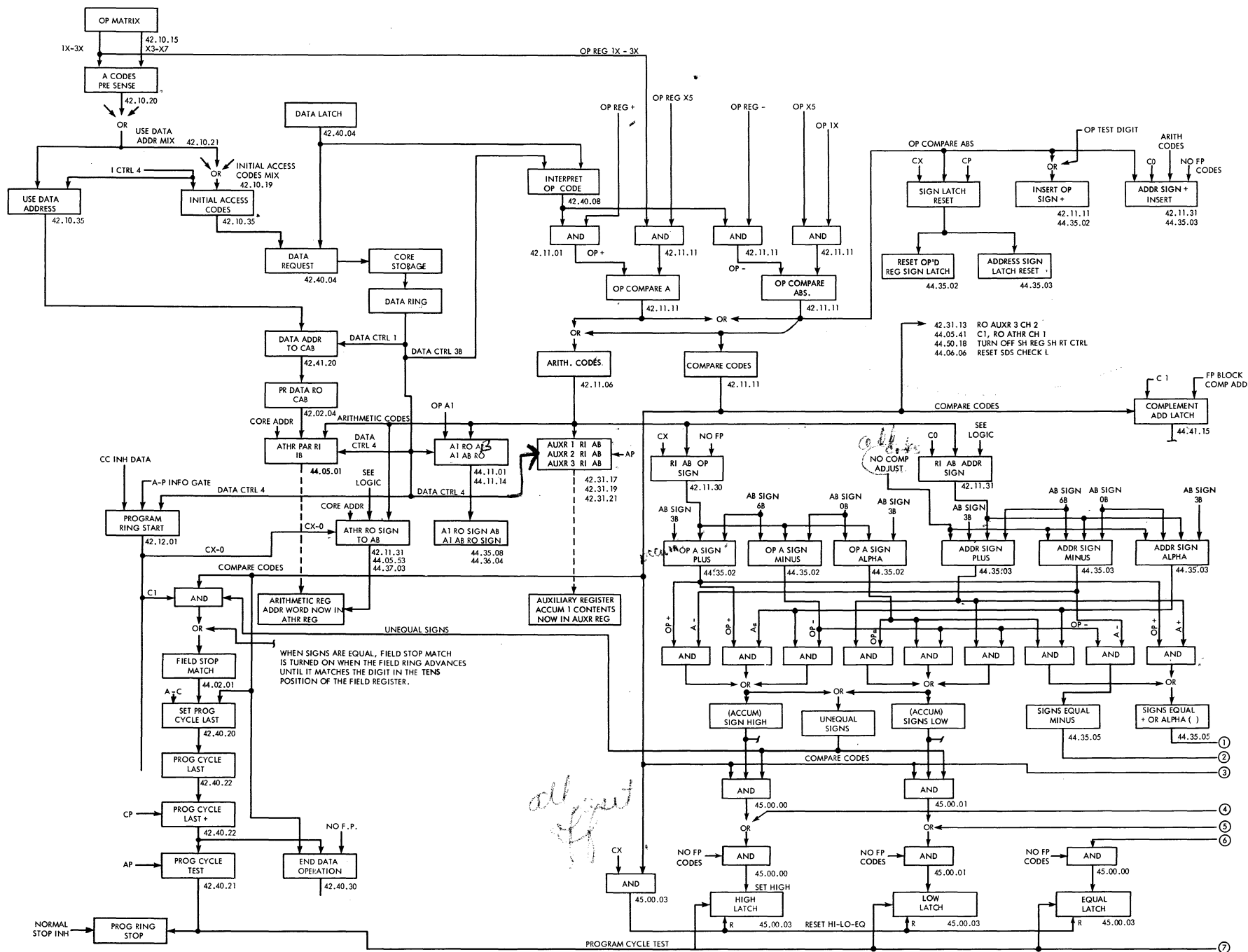


FIGURE 5.6-2A. COMPARE ACCUMULATOR AND COMPARE ABSOLUTE

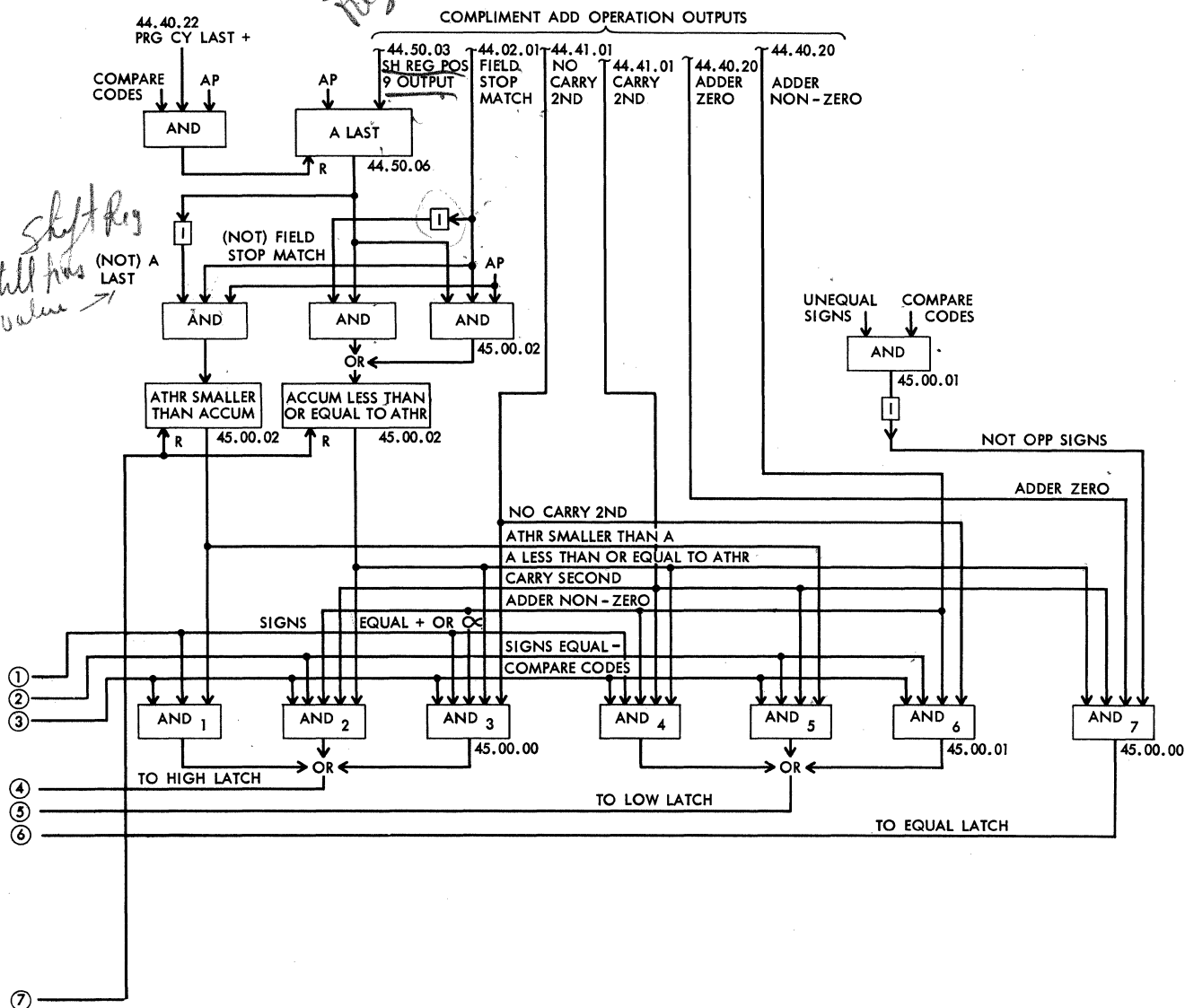


FIGURE 5.6-2B. COMPARE ACCUMULATOR AND COMPARE ABSOLUTE

Start Program Ring.

Gates Required:

A-P Info. Gate

"D" ctrl. 4

C. C. Inh Data

Turn on Op Sign and Adder Sign Latches

Gates Required:

A 1 RO Sign AB

"D" ctrl. 4

RI AB Op Sign

CX

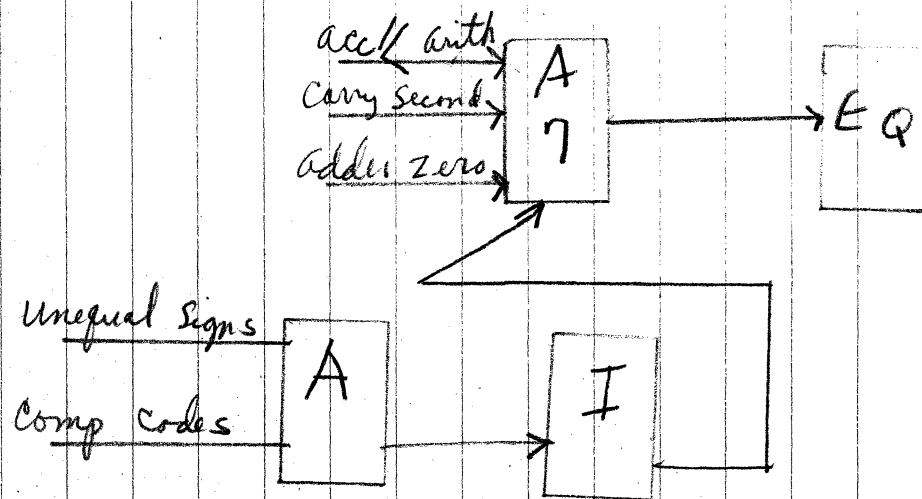
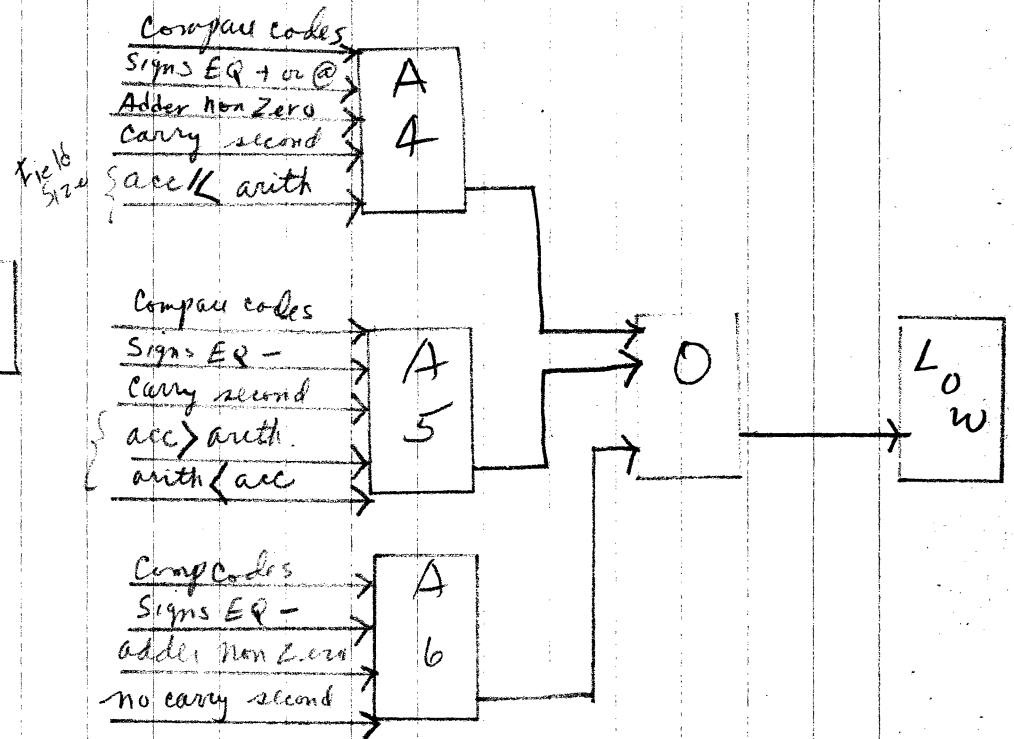
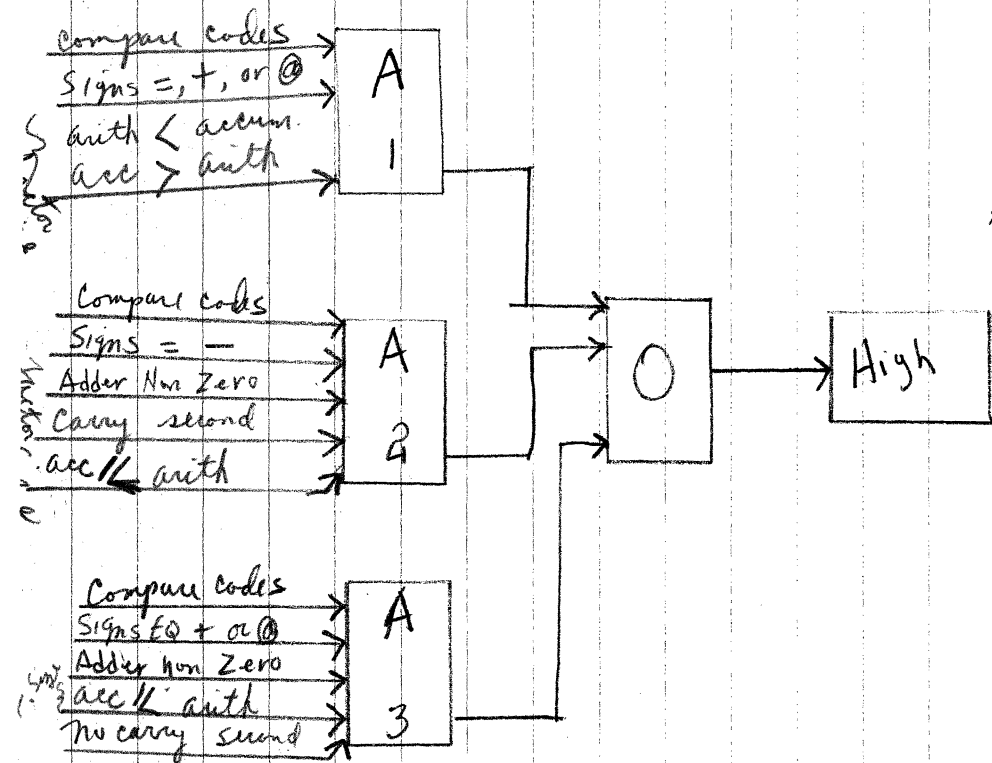
Athr RO Sign AB

CX-0

RI AB Adder Sign

C0

Set Hi, Low, or EQ indicators on Compare



The accumulator sign is read out to the arithmetic bus and then read in to the appropriate accumulator Op Sign latch at CX time.

The arithmetic register (core storage word) sign is read out to the arithmetic bus and then read in to the appropriate adder sign latch at C0 time.

Compare Signs. Figure 5.6-2 shows how the outputs of the op and adder sign latches are switched together to result in one of the following outputs:

1. Sign High
2. Sign Low
3. Signs Equal -, or Signs Equal + or Alpha

If the accumulator sign is either high or low, the appropriate high or low indicator latch is turned on as soon as the program cycle test gate is developed.

Signal Test Time if Unequal Signs. Whenever the sign test results in high or low accumulator sign, an unequal signal is developed which turns on field stop match at C1 time (Figure 5.6-4). This bypasses the usual circuit to turn on field stop match through the match of the field ring and the tens position of the field register. Field stop match turns on the program cycle ring. The program cycle test ring output switches with the result of the sign comparison to turn on either the high or low latch.

Signal Complement Add. If the signs are equal, field stop match cannot be turned on as just outlined, and the complement add operation continues. The compare codes signal and the output of the complement add latch provide the necessary gates to accomplish the complement add operation. The entire add operation is discussed in Section 5.1. The compare codes signal provides the gates to read out the arithmetic register to Channel 1 and the auxiliary register to Channel 2. Because Channel 2 goes to Adder Entry B and the true-complement switching, the accumulator contents are complemented.

Signal Test Time if Equal Signs. When the Field Ring advances until its value matches the digit in the tens position of the field register, a field stop match signal is developed. This causes the field ring to stop and starts the program cycle ring. When program cycle test comes on, the results of the complement add operation are tested to turn on either the high, low, or equal indicator latch.

Test for High, Low or Equal - Equal Signs. Figure 5.6-3 shows examples of the switching conditons resulting from the complement add comparison operation for a variety of factors in accumulator and core storage locations. Recall that the test takes place after the arithmetic register digits have passed through the adder and before the higher order auxiliary register digits have passed through the adder.

Switch 6 operates when the absolute value of the arithmetic register is less than the auxiliary register in the digit positions tested. Additional significant figures in the accumulator to the left of the positions tested cannot change the result. A Carry 2nd signal at switch 5 indicates that the absolute value of the Athr. is greater than the Auxr. in the digit positions tested. The addition of a Athr. Smaller than Accum. signal to the condition at switch 5 allows a switch output only when higher order accumulator digits exist. Figure 5.6-5 shows the sequence of operation.

End Data Operation and Stop Program Ring. As shown on Figure 5,6-2, the Last + output of the program cycle ring turns on end data operations and the test output causes a program ring stop.

Switch Number	(Address) Athr	(Accum) Auxr	Signals Developed	Indicator Latch Set	Machine Oper Example
1.	+ 695 ↙ 695	+ 7335 ↙ 7335	Athr Smaller than Accumulator Signs Equal + or Alpha	High	00695 92685 (c) ← 360
2.	- 832	- 695	A less than or equal to Athr Signs Equal - Adder non-zero Carry 2nd	High	00832 92305 (c) ← 137
3.	+ 695 ↙ 695	+ 832 ↙ 832	A less than or equal to Athr Signs Equal + or Alpha Adder non-zero No Carry 2nd	High	00695 92188 (n/c) ← 363
4.	+ 832 ↙ 832	+ 695 ↙ 695	A less than or equal to Athr Signs Equal + or Alpha Adder non-zero Carry 2nd	Low	00832 92305 (c) ← 137
5.	- 832	- 8831	Athr smaller than Accumulator Signs Equal - Carry 2nd	Low	00832 91169 (c) ← 001
6.	- 832	- 833	Signs Equal - Adder non-zero No Carry 2nd	Low	00832 92167 (n/c) ← 699
7. See Figure 5.7-1 for above Switches	+, -, ↙ 832	+, -, ↙ 832	A less than or equal to Athr Not opposite signs Adder zero Carry 2nd	Equal	00832 92168 (c) ← 000 Dotted line indicates test time.

FIGURE 5.6-3. SWITCHING CONDITIONS FOR COMPARE ACCUMULATOR

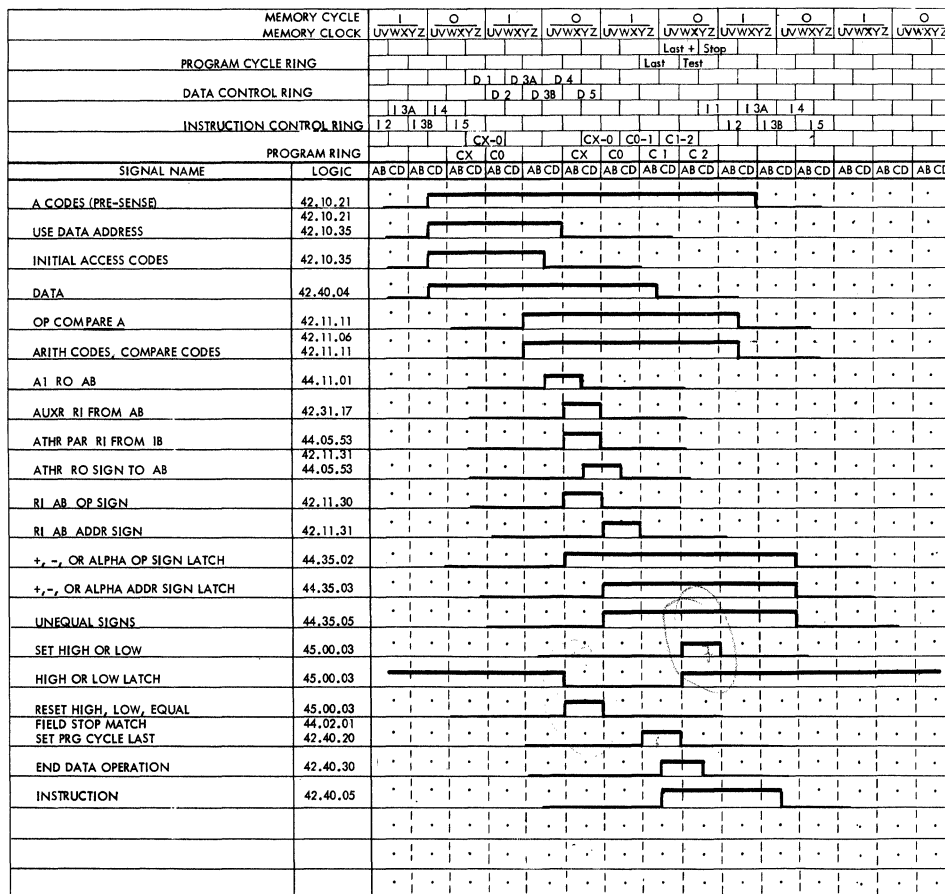


FIGURE 5.6-4. SEQUENCE FOR COMPARE ACCUMULATOR TO STORAGE, OP + 15, 25, 35, SIGNS NOT EQUAL

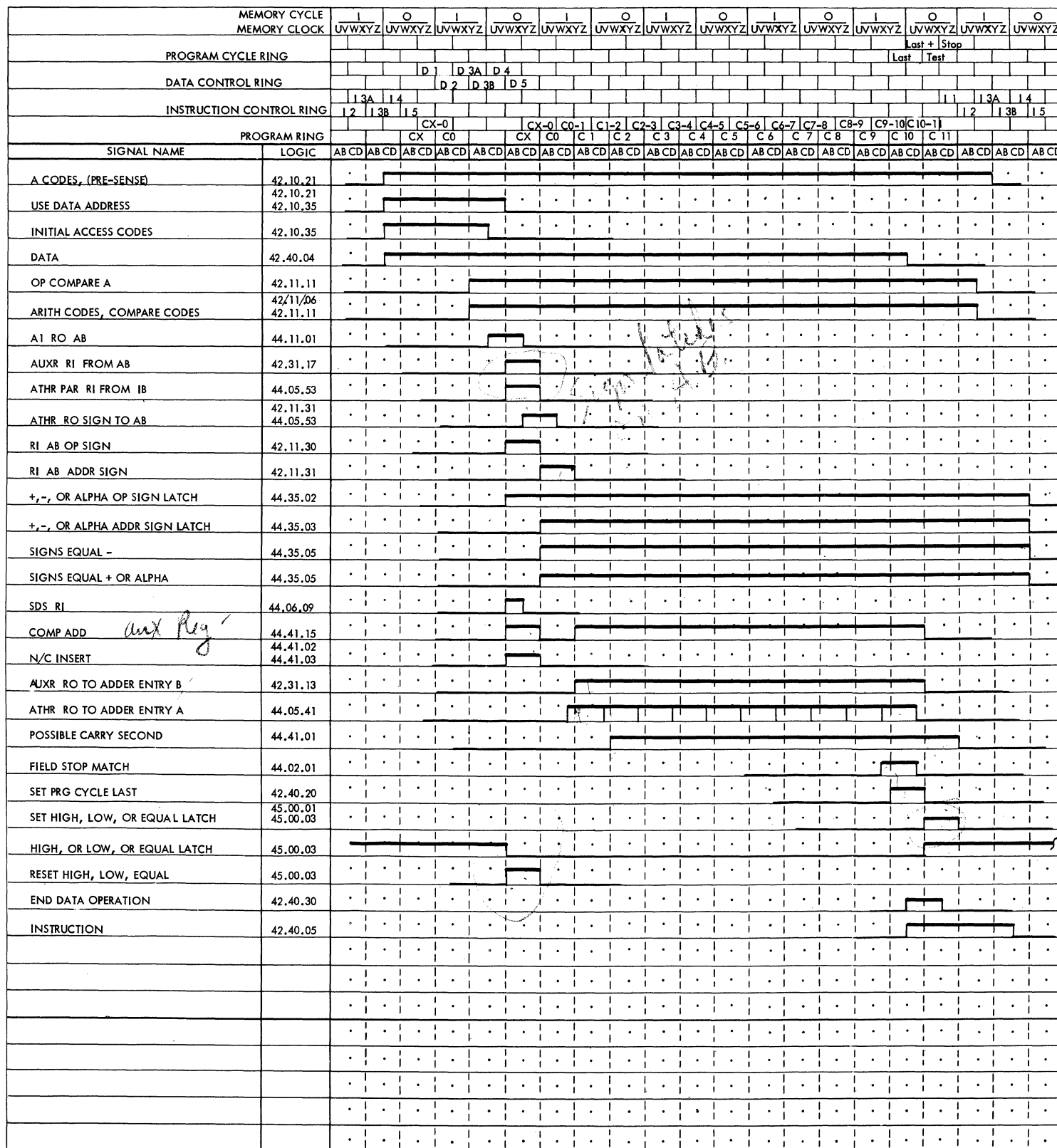


FIGURE 5.6-5. SEQUENCE FOR COMPARE ACCUMULATOR TO STORAGE,
OP + 15, 25, 35, FIELD EQUAL AND SAME SIGNS

5.6.02 Compare Absolute

Op - 15 (CA) Compare Absolute in Accumulator 1 to Absolute in Storage

The contents of Accumulator 1, considered plus, are compared with the field defined portion of the specified storage word, considered plus. If the accumulator contents are low, the low indicator latch is turned on; if high or equal, the corresponding latch is set on.

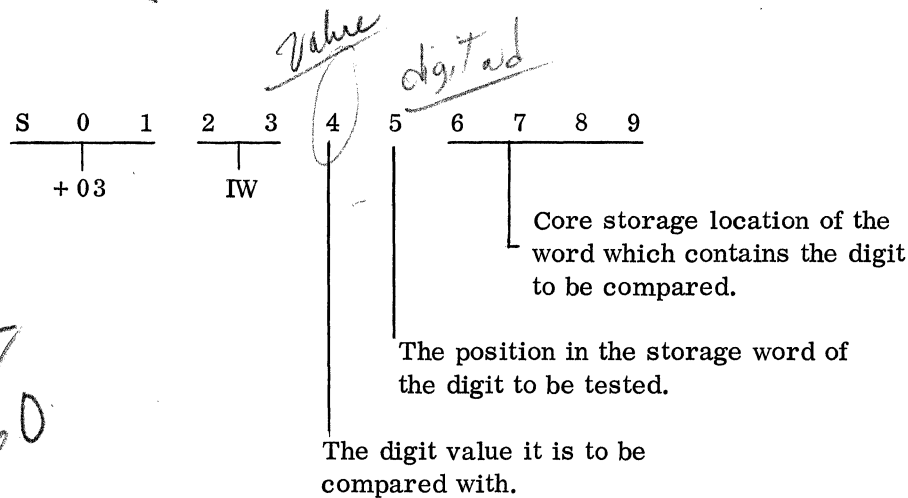
Circuits for Compare Absolute

This operation code uses the same circuits as compare accumulator # to Storage except that the Op'd Register Sign and Address sign latches are reset and plus signs are inserted in each latch. The development of the signals for Op Compare Absolute, Sign latch reset, and plus sign insert is shown in Figure 5.6-2. With both signs plus, the high, or equal, or low indicator is turned on as a result of the adder operation comparison.

5.6.03 Compare Storage to Digit

Op + 03 (CD) Compare Storage to Digit

This operation code compares the digit in the tens position of the field register with any designated digit in storage or in one of the accumulators. The sign of the storage word is not considered. The instructions format is used in the following manner:



Introduction

This is an initial access code in which the designated storage word is read out to the arithmetic register. The field register designated digit in the arithmetic register is then read out to the Adder B Entry where it is complemented. At the same time, the digit in the tens position of the field register is read out to Adder Entry A. A carry insert to the adder completes the gates needed for tens complement subtraction. The adder output is analyzed to determine the relative values of the two digits in accordance with the following chart.

Digit in Storage (Athr)	7	7	8
Digit in TP Fld Reg	7	9	6
Adder B Entry (Athr) (10's Compl)	3	3	2
Adder A Entry	(1) $\frac{7}{0}$	(1) $\frac{9}{2}$	(N/C) $\frac{6}{8}$
A Adder Output	Carry Adder Zero	Carry Adder non-Zero	No Carry Adder non-Zero
Compare Indicator latch	EQUAL	LOW	HIGH

Circuits for Compare Storage to Digit

Figure 5.6-6 outlines the circuits required for the compare digit operation. Figure 5.6-7 shows the sequence of operation. The following objectives are necessary to accomplish the comparison:

Read In to the Athr the Word Containing the Digit to be Compared. The usual initial access code data request is initiated. If the D-address in the program register is a core storage address, an Athr RI from IB gate results. If the D-address is an accumulator address, the designated accumulator is read out to the arithmetic bus and to the arithmetic register.

Insert Op'd Reg and Address Reg Plus Signs. A plus sign is inserted, at C0 time in both Op'd Register Sign and Address Sign latches so that both digits to be compared have + signs, regardless of the sign of the word of which they are a part (Figure 5.6-6).

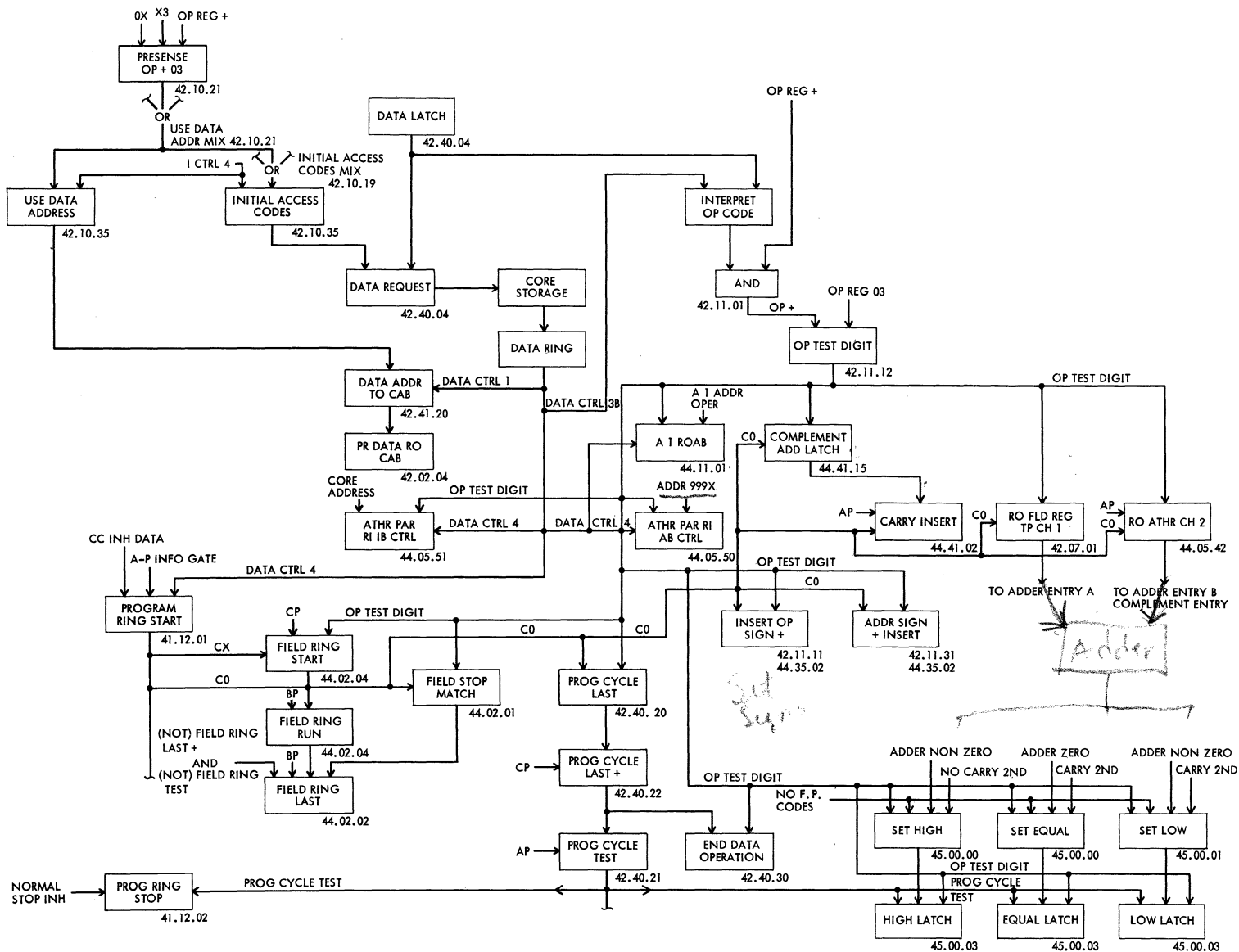
Read Out the Digit to be Compared From the Arithmetic Register. Field ring start, turned on at CX, CP time, switches with the units position output of the field register (Logic 44.03.01) to turn on the field register designated position of the field ring. The field ring output scans the output of the arithmetic register (Logic 44.01.11) to the Channel 2 mix (44.46.01). The RO Athr Ch 2 gate at C0 time, Figure 5.6-6, gates the selected digit from the arithmetic register to Channel 2.

Read Out the Field Register TP to Channel 1. Figure 5.6-6 shows the manner in which the R. O. Fld. Reg. TP Ch 1 signal is developed at C0 time.

Subtract the Athr Digit From the Fld. Reg. TP Digit. The complement add latch output controls the gates required for the subtract operation including the carry insert, Figure 5.6-6. See Section 5.1 for details of the arithmetic operation.

Test the Adder Output for High, Low or Equal. The conditions required at the adder output to turn on each indicator latch are discussed in the introduction.

FIGURE 5.6-6. COMPARE STORAGE TO DIGIT



MEMORY CYCLE		1		0		1		0		1		0		1		0		1		0	
MEMORY CLOCK		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ		UVWXYZ	
DATA CTRL., RING, INST CTRL.		D 4		D 5		I 1		I 2		I 3A		I 3B									
RING		+		+		+		+		+		+		+		+		+		+	
FIELD RING		Last		Test		Last		Test		Last		Test		Last		Test		Last		Test	
PROGRAM CYCLE RING		X-0		X-0		X-0		X-0		X-0		X-0		X-0		X-0		X-0		X-0	
PROGRAM RING		X		0		1		1		1		1		1		1		1		1	
SIGNAL NAME	LOGIC	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD	A8 CD
OP TEST DIGIT	42.11.12																				
CORE ADDRESS	42.05.05																				
ATHR PAR RI IB CTRL	44.05.51																				
COMPLEMENT ADD	44.41.15																				
CARRY INSERT	44.41.02																				
RO FLD REG TP TO CH1	42.07.01																				
RO ATHR CH 2	44.05.42																				
INSERT OP SIGN +	42.11.11 44.35.02																				
ADDR SIGN + INSERT	42.11.31 44.35.02																				
FIELD RING START	44.02.04																				
FIELD RING RUN	44.02.04																				
RESET HIGH, LOW, EQUAL	45.00.03																				
SET HIGH, LOW, EQUAL	45.00.03																				
END DATA OPERATION	42.40.30																				
PROG RING STOP	41.12.02																				
INSTRUCTION LATCH	42.40.05																				

FIGURE 5.6-7. COMPARE STORAGE TO DIGIT SEQUENCE

5.7.00 SIGN CONTROL, CODE - 03

This is an augmented code which is used for four different purposes. The digit in position 5 of the instruction word causes the following to occur:

Compare Signs (Operation 0)

The sign of the word specified by positions 6 - 9 of the instruction is compared to the sign indicated by position 4 of the instruction word. The sign coding for position 4 is shown in the following example. The high, low, or equal indicator latch is turned on, depending upon whether the sign of the addressed word is higher, lower, or equal to the position 4 sign.

Make (Set) Sign (Operation 1)

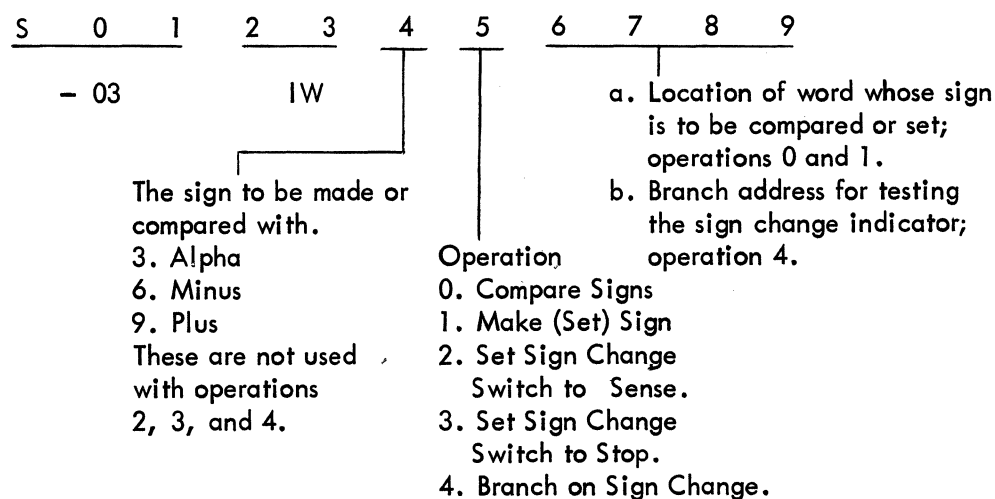
The sign of the word specified by positions 6 - 9 of the instruction is set plus, minus, or alpha as specified by the coded sign in position 4.

Set Sign Change Switch (Operation 2 or 3)

If the digit in position 5 is a 2, the sign change sense stop switch is set to sense; if a 3, to stop.

Branch on Sign Change (Operation 4)

Test the sign change latch and branch if it is on. If the latch is one, the test automatically turns it off.



The following six operation can be performed by a combination of the sign designation in position 4 (3, 6, or 9) with compare (0) or make (1) in position 5.

Position	4	5		
	3	0	Compare Sign to Alpha	CSA
	6	0	Compare Sign to Minus	CSM
	9	0	Compare Sign to Plus	CSP
	3	1	Make Sign Alpha	MSA
	6	1	Make Sign Minus	MSM
	9	1	Make Sign Plus	MSP

Note that a sign can be fully tested by the compare-sign-to-minus operation; plus sets high, minus sets equal and alpha sets low.

Introduction

Compare and Make. Operations 0 and 1 are both initial access codes, requiring that the word specified by the address portion of the instruction be read into the arithmetic register.

In the case of compare, the sign of the word in the Athr is transferred to the Op Sign latches, and the sign indicated by position 4 is read into the Address Sign latches. The high, low, equal sign compare circuits, described in section 5.6.01, are used to compare the signs. The high, low, or equal latch is turned on as a result of the comparison.

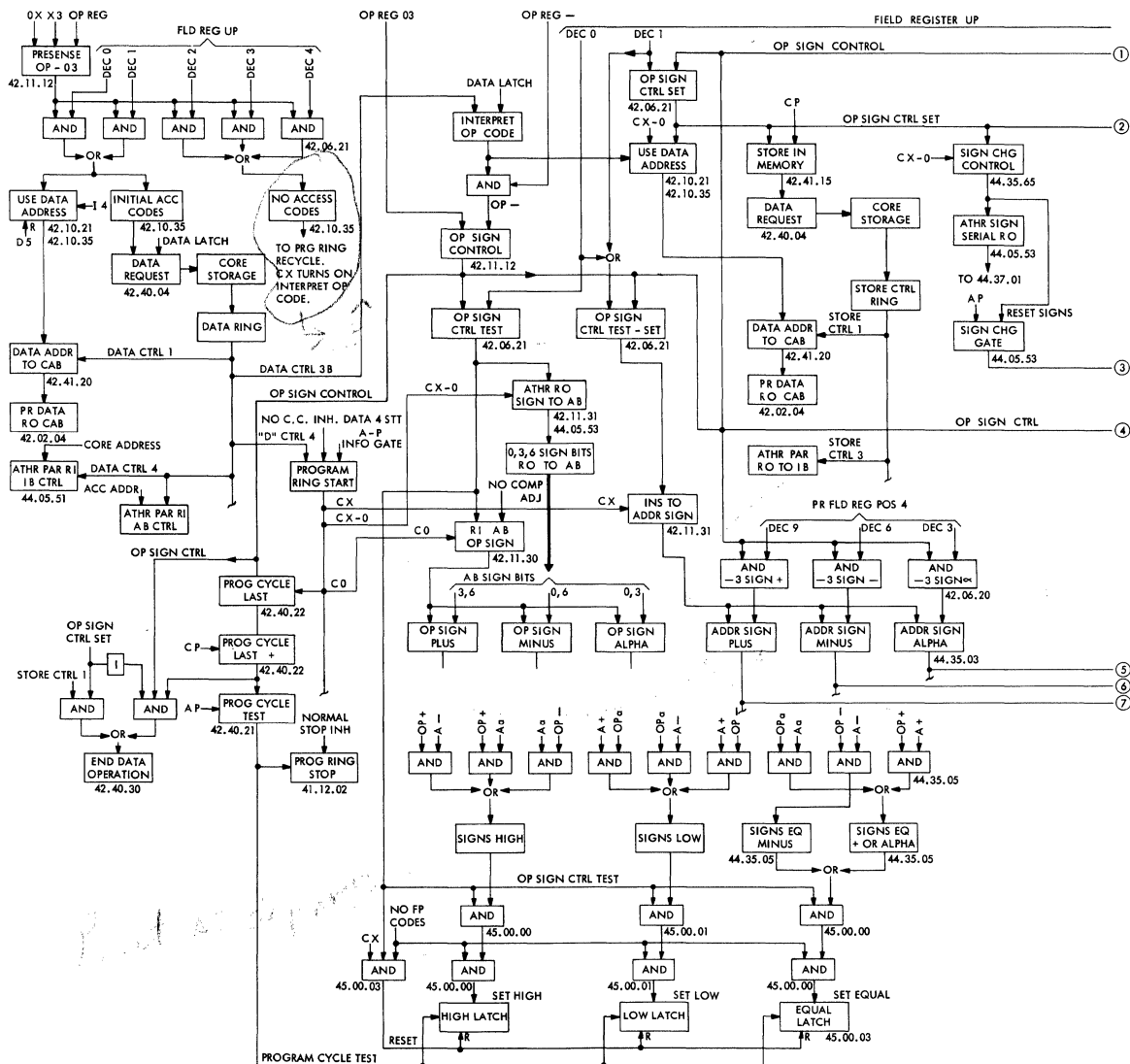


FIGURE 5.7-1A. SIGN CONTROL

In the case of make, Operation 1; the sign to be set, indicated by 3, 6 or 9 in position 4 of the instruction, is transferred from position 4 to the appropriate address sign latch. The sign in the arithmetic register is blanked on reset. The sign is now transferred from the address sign latch to the corresponding Athr sign core. The word in the arithmetic register must be returned to storage, so a store-in-memory operation is initiated to return the modified sign word to the storage location designated by the data address of the instruction.

Set Sign Change and Branch on Sign Change. Operations 2, 3 and 4 are No Access Codes, requiring that the program ring be recycled to turn on the Interpret Op Code signal.

If position 5 of the program register is a 2, the sign sense latch is set to Sense; if position 5 is a 3, the sign sense latch is set to Stop. A test is made, following the switch set, to determine whether it was set to correspond with the indicated operation, 2 or 3. A Branch-No-Branch Error Signal results if the sign sense latch is not correctly set.

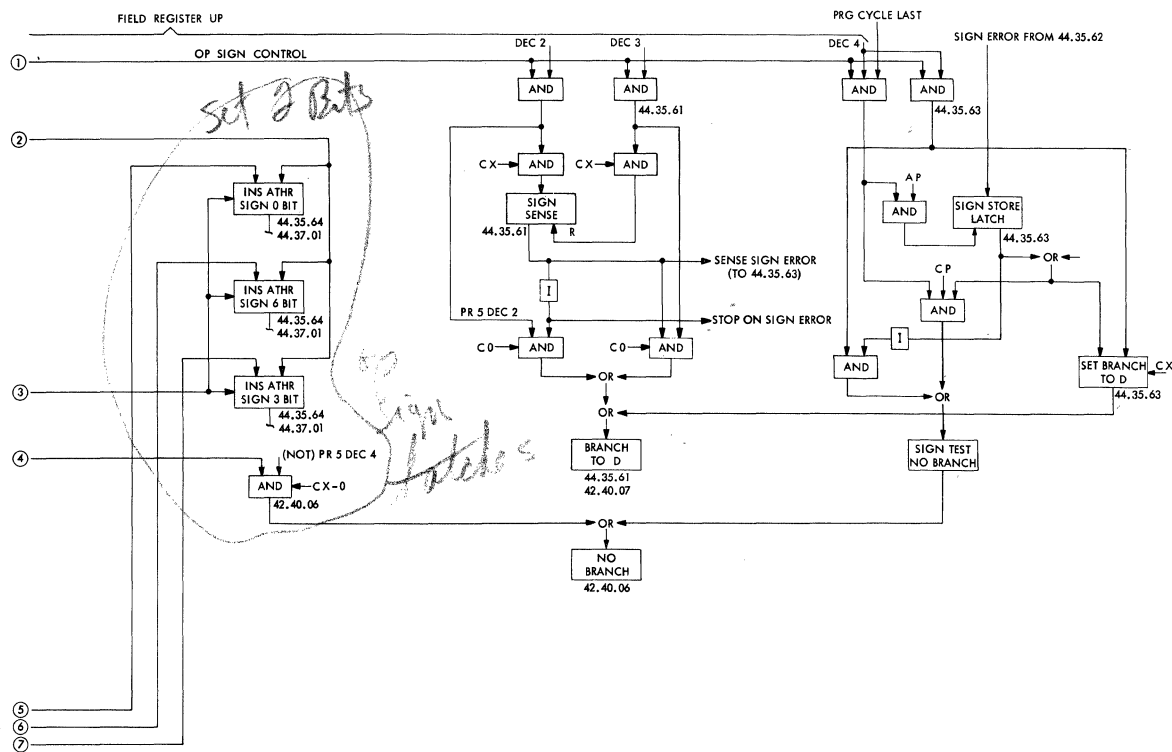


FIGURE 5.7-1B. SIGN CONTROL

If the operation is Branch on Sign Change, a Branch-to-D-signal is developed if the indicator latch is on; a No Branch, if off. The latch is reset if on, and a test is made to determine whether it was reset. Failure to reset causes a Branch, as well as No Branch with a resulting Branch-No Branch Error.

Circuits for Sign Control Figure 5.7-1

Compare Signs. The following objectives are accomplished to compare the signs of the designated word in storage with the sign indicated by position 4 of the instruction. Figure 5.7-2 shows a sequence chart of the operation.

1. Presense the Op Code

Figure 5.7-1 shows how the -03 output of the Op Code register is switched with the field register units position output (operation) to signal initial access codes and use-data address.

2. Read-out the word whose sign is to be compared

The initial access codes signal, developed when the Field Reg UP is Decimal 0, causes a conventional data request operation. The word whose sign is to be compared is brought to the arithmetic register.

3. Sense the Op Code; Sign Control, Operation Compare

As shown in Figure 5.7-1, the Data Control 3B output switches with a data cycle signal to cause Interpret Op Code. This in turn switches with the Op Register output and a Fld. Reg Decimal 0, Compare signal to cause the following:

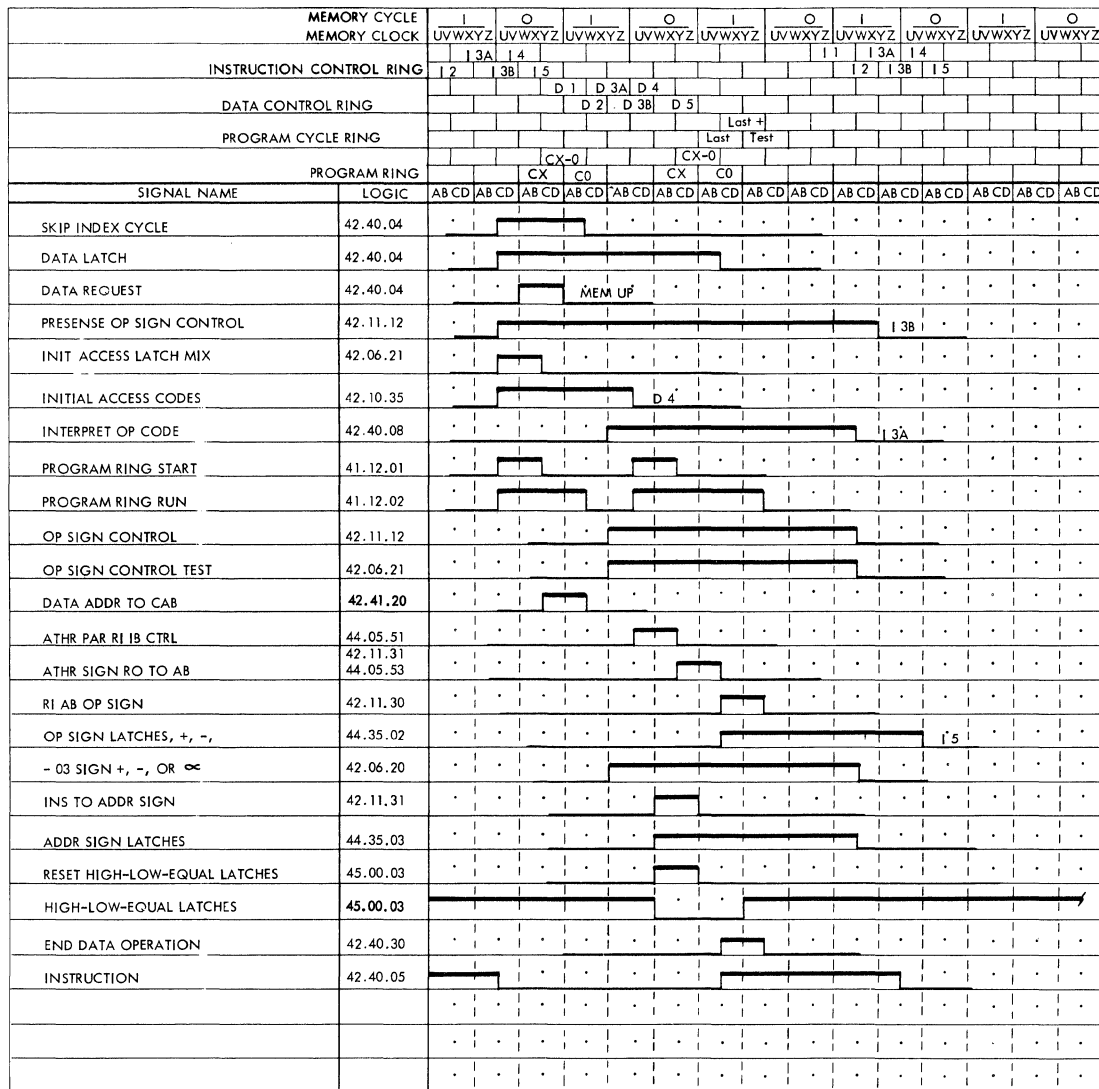


FIGURE 5.7-2. SIGN CONTROL, COMPARE (-03 00X0 XXX)

- a. Op Sign Control
 - b. Op Sign Control Test
 - c. Op Sign Control Test-Set
4. Read Out the Athr Sign to the Op Sign latches

Figure 5.7-1 shows how the Athr Sign is read out to the arithmetic bus at CX-0 time. At C0 time, a RI AB Op Sign gate causes the sign bits on the AB to turn on the appropriate Op Sign latch.

5. Read Out the Sign from the PR Fld. Pos 4 to the Addr Sign Latches

The Op Sign Ctrl Test-Set signal develops a CX, Ins. to Addr Sign gate which samples the PR Field Position 4 output to turn on the corresponding address sign latch.

6. Compare Signs and Set High, Low, or Equal

With one op sign latch and one address sign latch on, the sign compare switching (See section 5.6.01) causes either set high, set low, or set equal. The Set signal is sampled by a program cycle test gate to turn on either the high, low, or equal indicator latch.

7. Start the Program Cycle Ring

The program cycle ring is started by a program ring C0 gate as shown on Figure 5.7-1. The ring outputs accomplish the following:

- a. The program cycle last gate switches with Op Sign Ctrl to turn on end data operation except when the sign is being set.
- b. Program cycle test causes the program ring to stop and provides the test gate to set the high, low, or equal latches.

Make (Set) Signs. The following objectives must be fulfilled to set the sign of the designated word to correspond with the codes sign indicated by position 4 of the instruction. A sequence chart of the operation is shown in Figure 5.7-3.

1. Pre-sense the Op Code. See Compare (Figure 5.7-1).
2. Read out the word whose sign is to be set. See Compare (Figure 5.7-1).
3. Sense the Op Code; Sign Control, Operation Set.

The following Op Signals are developed in the same manner as in the compare operation.

- a. Op Sign Control
 - b. Op Sign Control Test-Set
 - c. Op Sign Control Set
4. Place sign to be set in address sign latch

In the same manner as with Compare, the Op Sign Control Test-Set signal causes a CX time gate which samples the PR Fld Pos 4 output to turn on one of the address sign latches.

5. Reset previous Athr Sign and gate transfer of sign from Address Sign latch to Athr sign cores.

The Op Sign Control Set signal switches with a CX-0 gate to provide a sign change control gate which causes:

- a. An Athr Sign Serial RO gate which causes the reset of the sign cores.
- b. An AP time Sign Change Gate which samples the appropriate Address Sign latch output to drive one of the Sign core read-in circuits.

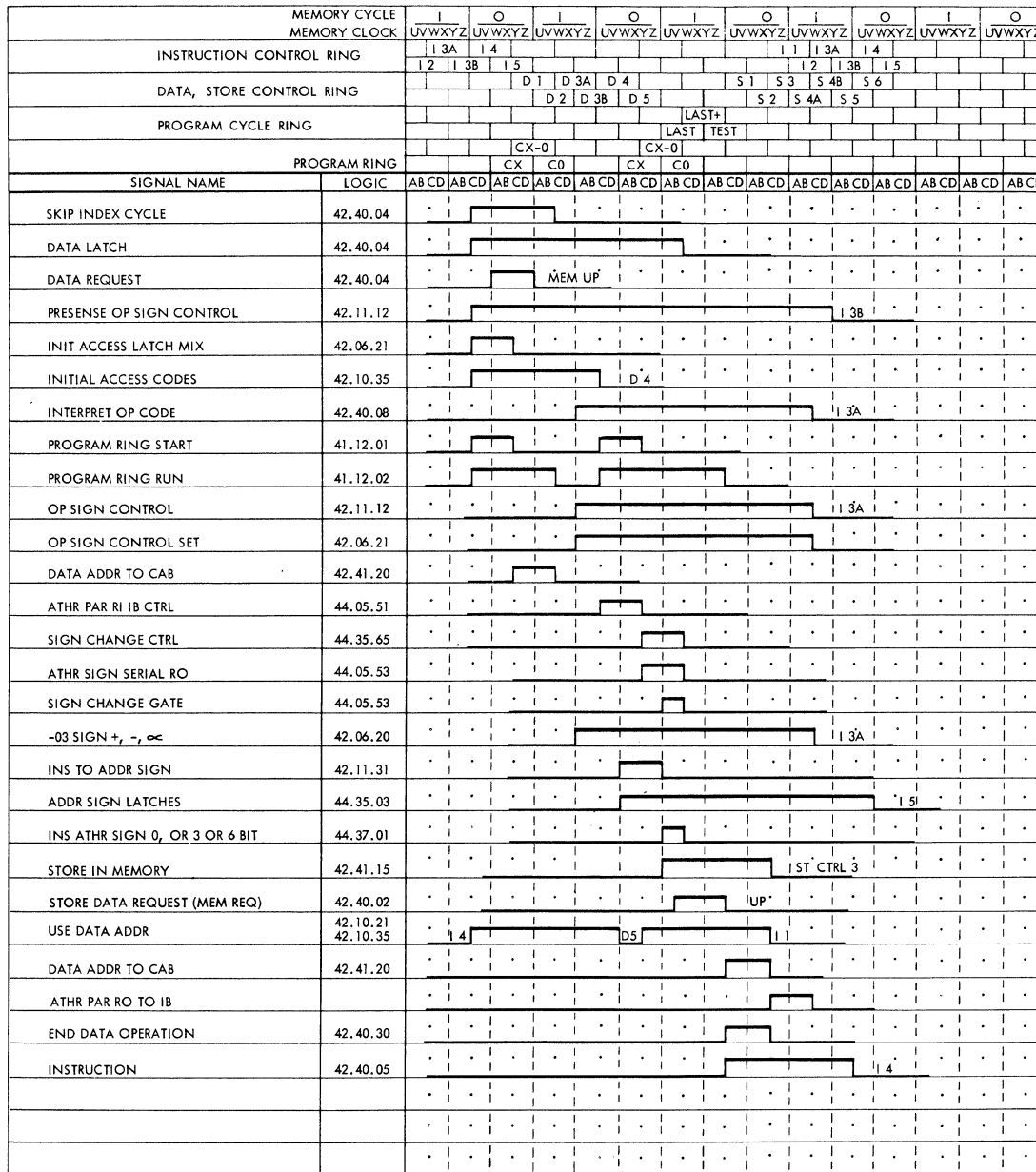


FIGURE 5.7-3. SIGN CONTROL, SET (-03 00 X1 XXX)

6. Initiate a Store in Memory Operations

The word in the Athr, with its newly set sign, is transferred back to memory by a conventional store-in-memory operation as shown in Figure 5.7-1. Note that the use-data-address signal is brought up again at CX-0 time for the Set Operation. Figure 5.7-3 shows how the signal, developed for the initial access to memory, drops at Data Control 5 time.

Set Sign Change Switch. Figure 5.7-4 shows the sequence of operation for the setting of the sign change switch. Figure 5.7-1 shows how PR Pos 5 Dec 2 switches with Op Sign Control, and CX to set the sign change switch to sense sign error. If the PR Pos 5 output is 3, the sign sense switch is reset so that the stop-on-sign error signal is up.

Immediately after the setting of the sign change switch, a check is made to be sure it was set to the correct position. At C0 time, PR 5 Dec 2 switches with the inverted sense-sign-error signal. If the switch is not set to Sense Sign Error, a Branch-to-D signal results. Because No Branch is always turned on, except on the PR 5 Dec 4 operation, (Figure 5.7-1) a Branch-No Branch Error signal is developed (Logic 42.40.06). A similar check is made when the switch is turned to the stop-on-sign-error position.

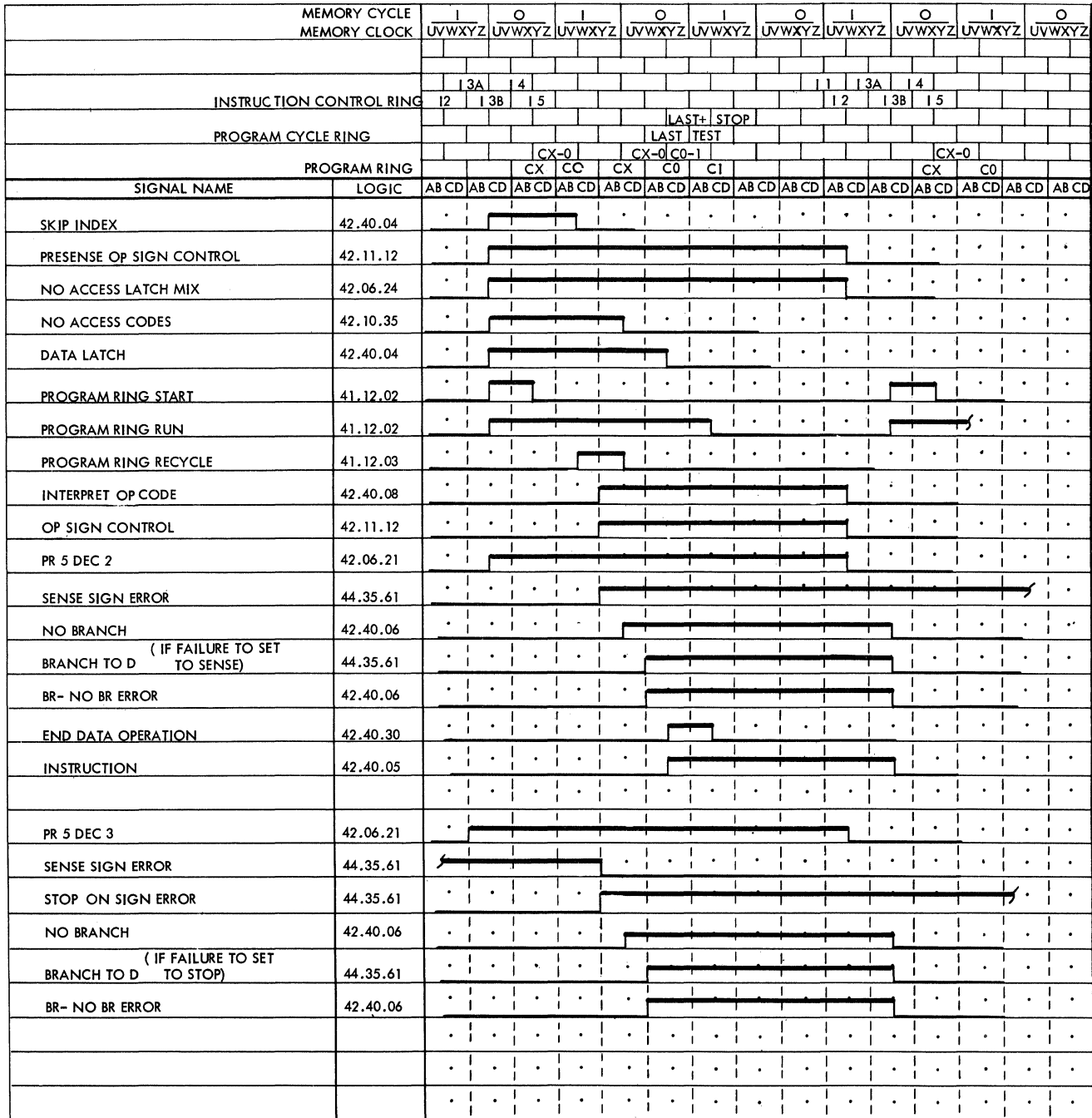


FIGURE 5.7-4. SIGN CONTROL, SET SIGN CHANGE SWITCH
(-03 00 02 XXX SET TO SENSE)
(-03 00 03 XXX SET TO STOP)

Branch On Sign Change. Figure 5.7-1 and 5.7-5 show how a CX time Branch to D signal develops when PR Dec 4, Op Sign Control signal switches with the on output of the sign store latch. If the latch is off, the PR Dec 4, Op Sign Control Signal switches with the inverted latch output to cause a No Branch.

When the latch is on, it is reset off by a Program Cycle Last AP. A test is then made at Program Cycle Last CP time to be sure the latch is off. If the sign store latch is not off, Sign Test No Branch is turned on. With Branch-to-D previously turned on, a Branch-No Branch Error will result.

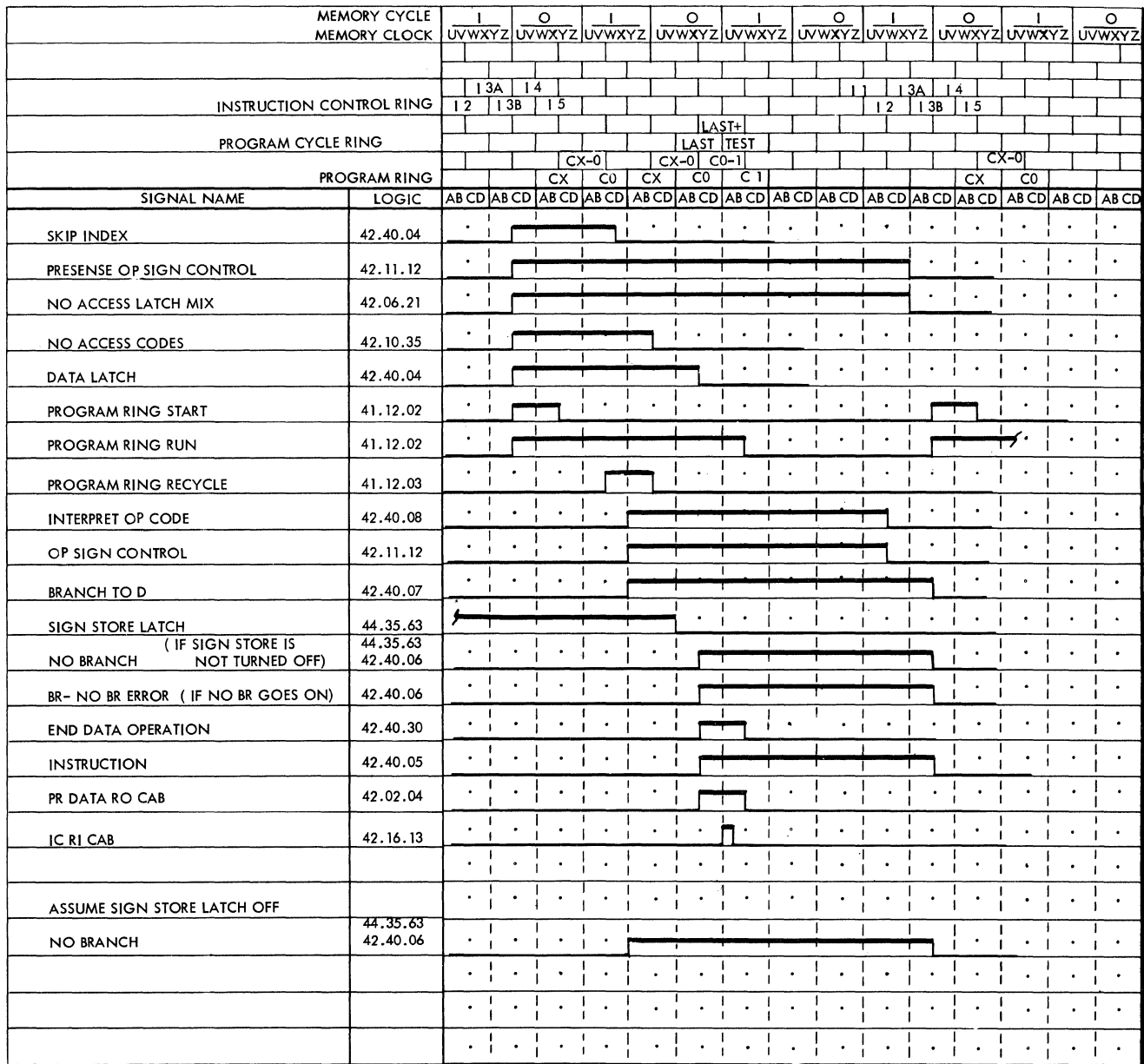


FIGURE 5.7-5. SIGN CONTROL, BRANCH (-03 00 04 XXXX)
SIGN STORE LATCH TURNED ON

5.8.00 BRANCH INSTRUCTIONS

Many of the branch codes described in the following material do not require access to memory and are termed "No-Access Codes." The program controls operate somewhat differently than for the Initial Access Codes described in section 4.

The block diagram of Figure 5.8-1 shows a typical program control operation for a no access type branch code. After the new instruction enters the program operation latch

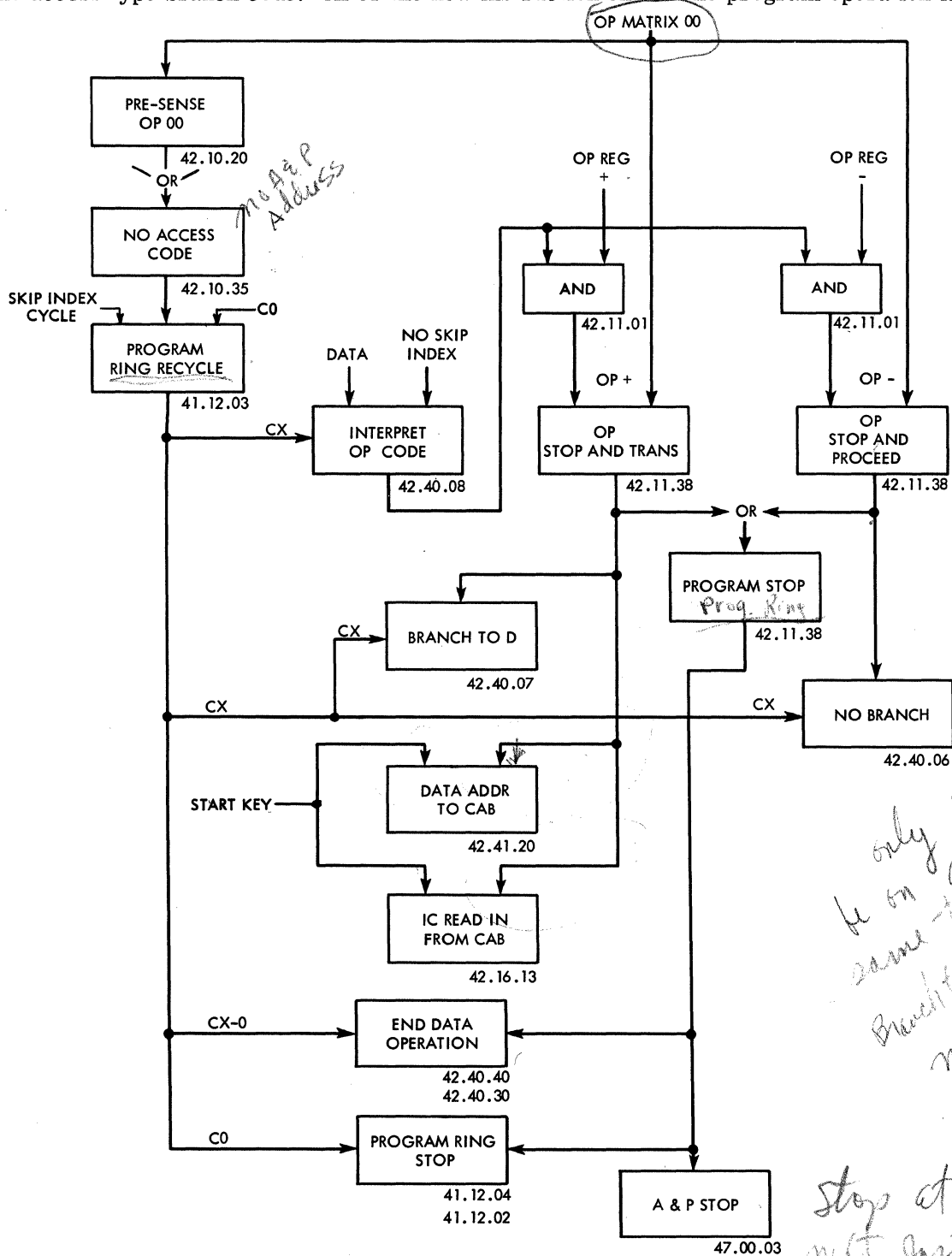


FIGURE 5.8-1. HALT AND BRANCH, HALT AND PROCEED

register during I Control 4 time, the register output is analyzed to create a pre-sense op line. This mixes with other No Access Codes to turn on a program ring recycle control signal at C0 time (Prog ring started by I Control 4). The skip index cycle version is shown. When an index cycle is taken, a similar action occurs at C5 during the index cycle (Figure 3.2-2). The program ring recycle latch (Figure 3.2-2) causes the stop of the program ring and a restart at the CX position.

The skip index cycle latch is turned off by a program ring recycle, DP (Figure 5.8-2). With skip index off, Data latch on, and CX (recycle), the interpret operation code latch can be turned on. The interpret op code signal switches with the sign output of the op register and, in turn, with the op matrix output to turn on the operation latches which are used to directly control the performance of the op code. The program ring outputs are used to time various functions including, in the case of codes ± 00 , the end data operation and program ring stop signals.

5.8.01 Halt and Proceed, and Halt and Branch

Op +00 (HB), Halt and Branch

This operation causes an unconditional machine stop. The console typewriter automatically types the contents of the instruction counter and the program register. When the start key is depressed, the next instruction is taken from the address in program register D.

Op - 00 (HP), Halt and Proceed

Halt and Proceed operates the same as Halt and Branch except that the next instruction is taken from the instruction counter rather than program register D.

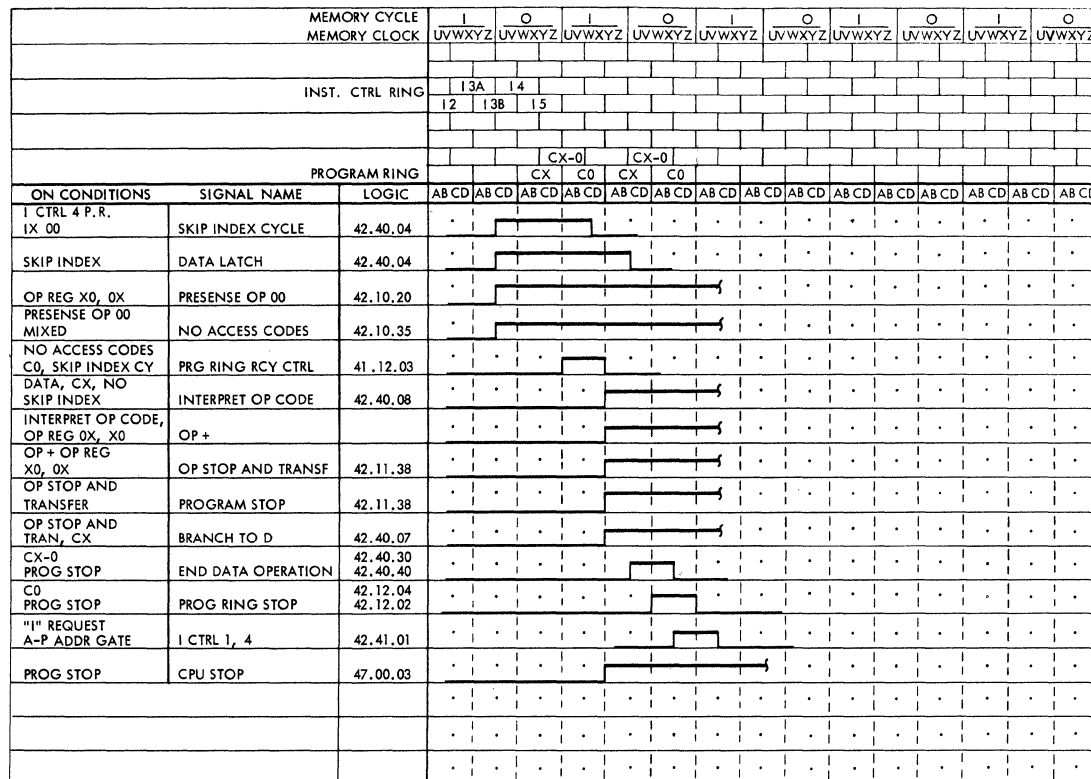


FIGURE 5.8-2. HALT AND BRANCH

Introduction

To stop the program, a program stop signal must be sent to the console to turn off the Run condition. This causes the Run line to go down (Figure 4.1-2) and prevents the instruction latch from being turned on by the normal end-of-data-operation signal. With the instruction latch off, the instruction control ring cannot be started. All data cycle control signals are dependent upon the I Control ring outputs for turn off pulses. Thus the various controls for both halt codes remain on until the start key is depressed and the program is restarted by a start gate turning on this Instruction latch. The Start signal, in the case of the Halt and Branch code, also causes the transfer of program register D contents to the instruction counter.

Circuits for Halt and Branch +00 (Figure 5.8-1, 2)

The Interpret Op code, +00 Op matrix signal development is shown in Figure 5.8-1. Both Op Stop and Transfer, and Program Stop are turned on by the op code analysis. Program stop turns on end-data operation which causes the data latch to go off. The instruction latch is not turned on by end-data operation because the program stop signal to the A&P Stop circuits causes the Run line to go down (Figure 4.1-2 and 5.8-2). As noted previously, the failure of the instruction control ring to run allows the same op code to remain in the op code register. Thus, both Op Stop and Transfer, and program stop remain up. When the start key is depressed, program register D is read out to the CAB, and the instruction counter is signalled to read in from the CAB.

Circuits to cause the console typewriter to automatically type the contents of the Instruction Counter and Program Register are discussed in the Console section.

Circuits for Halt and Proceed, -00 (Figure 5.8-1, 3)

The circuits for Halt and Proceed are very similar to those for Halt and Transfer. Figure 5.8-1 indicates the differences. After stopping, depression of the start key causes the instruction latch to come on (Figure 4.1-2) and the program continues using the instruction counter address for the next instruction location. Program register is not transferred to the instruction counter.

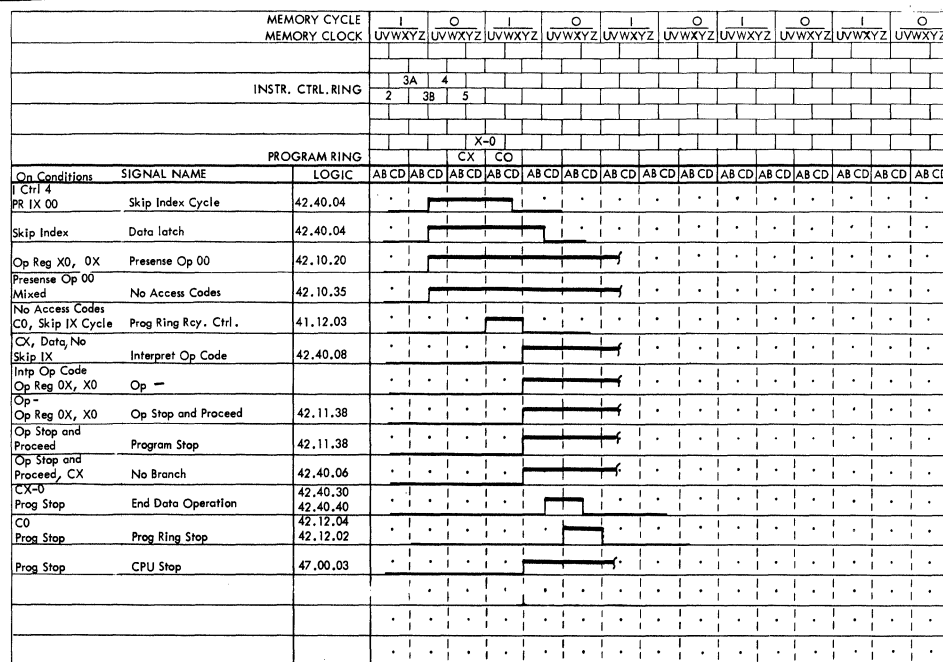


FIGURE 5.8-3. HALT AND PROCEED

5.8.02 Branch and No Op

Op +01 (B) Branch

This is an unconditional branch. The next instruction is taken from the address in program register D.

Op -01 (NOP) No Operation

No Operation is performed. The next instruction is taken from the address in the instruction counter.

Introduction

Both Branch and No Operation are No-Access codes which operate as described in Section 5.8.00. Figures 5.8-4, 5 show the no-access code program control operation following an index cycle. The operation is similar to that following a skip index cycle except that the program ring is recycled at C5 time instead of C0 time.

To branch to the address specified by program register D, the contents of the instruction counter must be reset, and the program register read out to the instruction counter (IC) via the computer address bus (CAB). During I Control 1 time of the next instruction cycle, the new IC address is read out to the core storage address register from program register D via the CAB.

Circuits for Branch, +01 (B)

When Interpret Op Code comes on (Figure 5.8-4), the Op + signal is developed. This causes the Op Transfer Unconditional signal to come on, which in turn mixes to turn on Op Branch Codes. Similarly, the Op Branch Unconditional signal comes up.

The Op Transfer Unconditional signal sets the program cycle last plus latch. This is the second step of a 6 step ring which is similar to the control rings. The ring may be started at any step, depending upon the operation being performed. The 6 steps are as follows:

1. Program Cycle Last -
2. Program Cycle Last
3. Program Cycle Last +
4. Program Cycle Test
5. Program Cycle Stop
6. Program Cycle Stop +

The Op Branch Unconditional signal turns on Branch to D at Program Ring CX-0 time. This results in the transfer of the contents of program register D to the instruction counter at Program Cycle Last + time.

The Program Cycle Last + gate also turns on the end-data operation latch which causes the reset of the data latch and the set of the instruction latch (Figure 4.1-2).

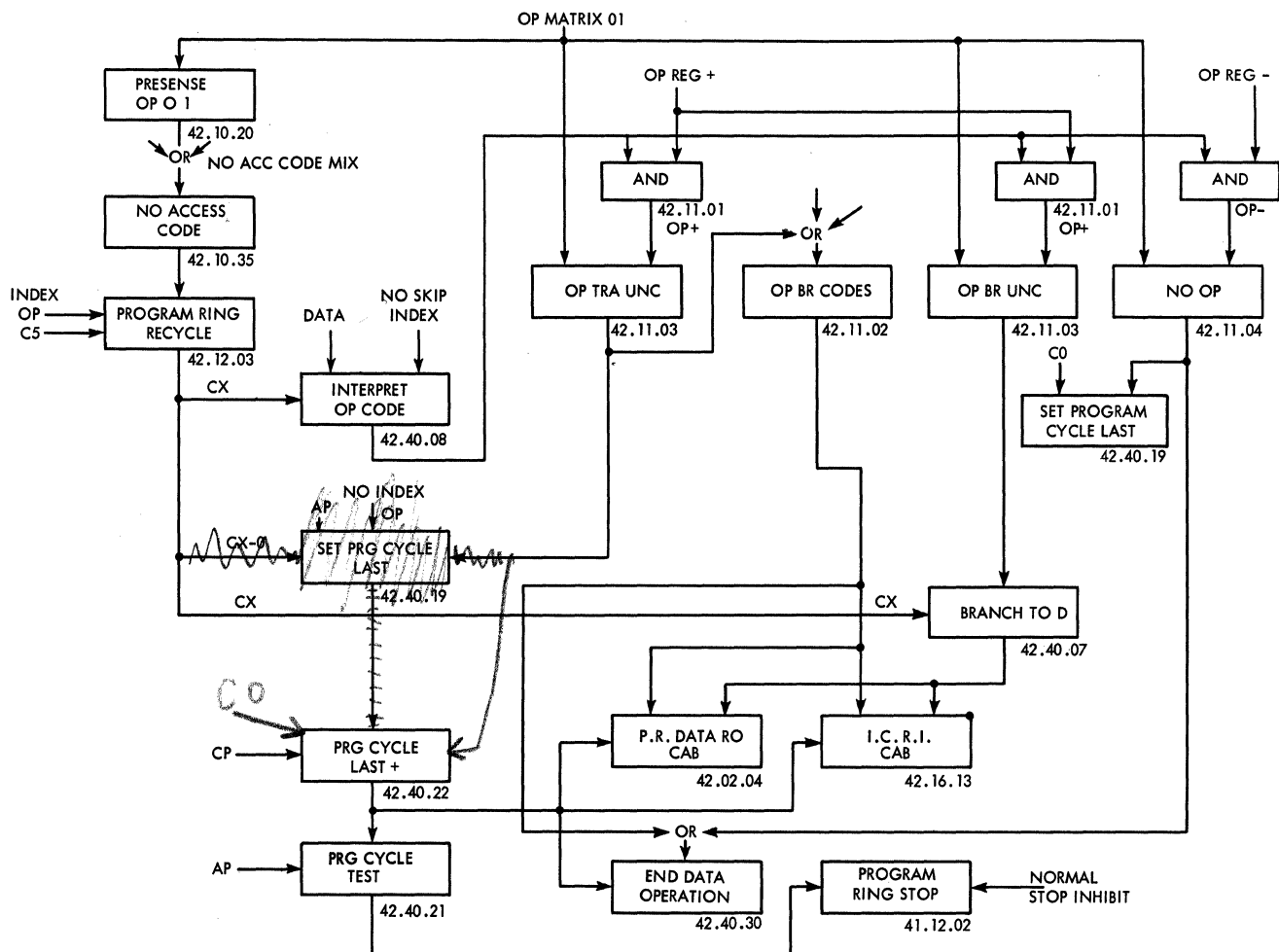


FIGURE 5.8-4. BRANCH AND NO OP

		MEMORY CYCLE MEMORY CLOCK																
		I				O				I				O				
		UVWXYZ				UVWXYZ				UVWXYZ				UVWXYZ				
INST CONTROL RING																		
		Last + Stop																
PROGRAM CYCLE RING		Test																
PROGRAM RING		C3-4				C4-5				CX-0								
SIGNAL NAME		LOGIC				C3	C4	C5	CX	C0	AB	CD	AB	CD	AB	CD	AB	CD
INDEX OPERATION	42.40.09					.	.	.	.	.	.	.	.	.	.	.	.	.
PROGRAM RING RECYCLE CONTROL	41.12.03					.	.	.	.	.	.	.	.	.	.	.	.	.
PROGRAM RING RECYCLE	41.12.03					.	.	.	.	.	.	.	.	.	.	.	.	.
PROGRAM RING STOP	41.12.02					.	.	.	.	.	.	.	.	.	.	.	.	.
RING RUN	41.12.02					.	.	.	.	.	.	.	.	.	.	.	.	.
DATA LATCH	42.40.04					.	.	.	.	.	.	.	.	.	.	.	.	.
INTERPRET OP CODE	42.40.08					.	.	.	.	.	.	.	.	.	.	.	.	.
OP TRA UNC (+01)	42.11.03					.	.	.	.	.	.	.	.	.	.	.	.	.
OP BR UNC	42.11.03					.	.	.	.	.	.	.	.	.	.	.	.	.
OP BR CODES	42.11.02					.	.	.	.	.	.	.	.	.	.	.	.	.
BRANCH TO D	42.40.07					.	.	.	.	.	.	.	.	.	.	.	.	.
SET PROGRAM CYCLE LAST +	42.40.19					.	.	.	.	.	.	.	.	.	.	.	.	.
PR DATA RO CAB	42.02.04					.	.	.	.	.	.	.	.	.	.	.	.	.
IC RI CAB	42.16.13					.	.	.	.	.	.	.	.	.	.	.	.	.
END DATA OPERATION	42.40.30					.	.	.	.	.	.	.	.	.	.	.	.	.
INSTRUCTIONS	42.40.05					.	.	.	.	.	.	.	.	.	.	.	.	.
INSTRUCTION REQUEST	42.40.05					.	.	.	.	.	.	.	.	.	.	.	.	.
						.	.	.	.	.	.	.	.	.	.	.	.	
						.	.	.	.	.	.	.	.	.	.	.	.	
						.	.	.	.	.	.	.	.	.	.	.	.	
						.	.	.	.	.	.	.	.	.	.	.	.	

FIGURE 5.8-5. SEQUENCE FOR BRANCH (B)

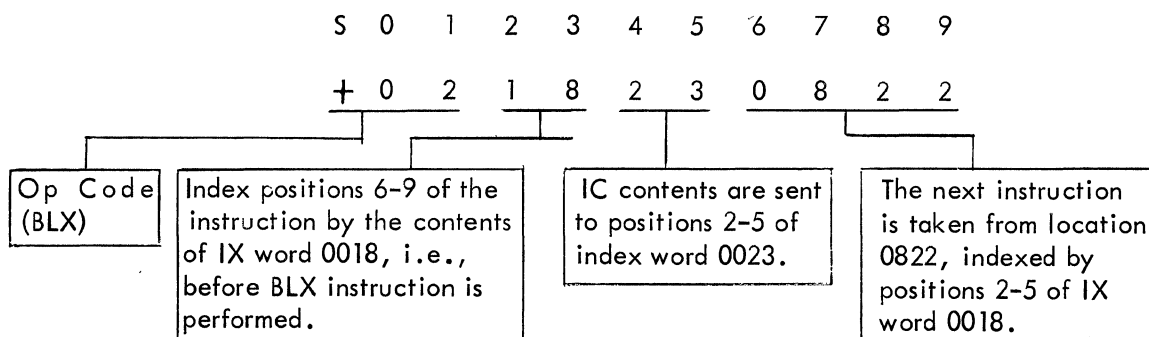
Circuits for No Op, -01 (NOP)

The no-operation signal is developed when the Op Register is minus (Figure 5.8-4). It operates in the same way as Branch, except that Branch to D and the ensuing PR Data to IC transfer are not developed. The Program Cycle Last + signal turns on end-data operation, and the next instruction is taken from the instruction counter.

5.8.03 Branch and Load Location in Index Word

Op + O2 (BLX)

This operation causes the contents of the instruction counter to be sent to positions 2 - 5 of the index word specified by positions 4 - 5 of the instruction. The next instruction is taken from the address in program register D. The D address can be indexed normally by positions 2 - 3 of the instruction.



Introduction

The performance of this operation code can be divided into three major objectives as follows:

Index Word to Auxiliary Register. Because the op code requires access to the index word in core storage, it is an initial access code. Section 4 describes the program controls for initial access codes. A data request is immediately generated which results in the address in positions 4 - 5 being sent to the core storage address register. Zeros are inserted in the hundreds and thousands position (address 0023 in example above). The contents of the selected index word are read into the auxiliary register via the information bus.

Insert I. C. Contents in IX Word and Store IX Word in Memory. The insertion of the instruction counter contents in the index word is accomplished by reading out the instruction counter to the information bus (positions 2 - 5) at the same time that auxiliary registers I and III (positions SO-1 and 6 - 9) are being read out to the information bus. The read out of both auxiliary registers I and III and the instruction counter take place during a store-in-memory operation in which the revised index word is returned to core storage (location 0023 in the example).

Circuits for Branch and Load Location in Index Word

FIGURE 5.8-6. BRANCH AND LOAD LOCATION IN INDEX WORD

The Op Code signal is developed in the usual way by switching the Op Matrix output with Interpret Op code. The Op +02 signal then turns on Op Store IC Transfer which causes the storage of the index word back in the location specified by the field address. The Op +02 signal also turns on Op Branch Unconditional which causes the contents of program register D to transfer to the instruction counter.

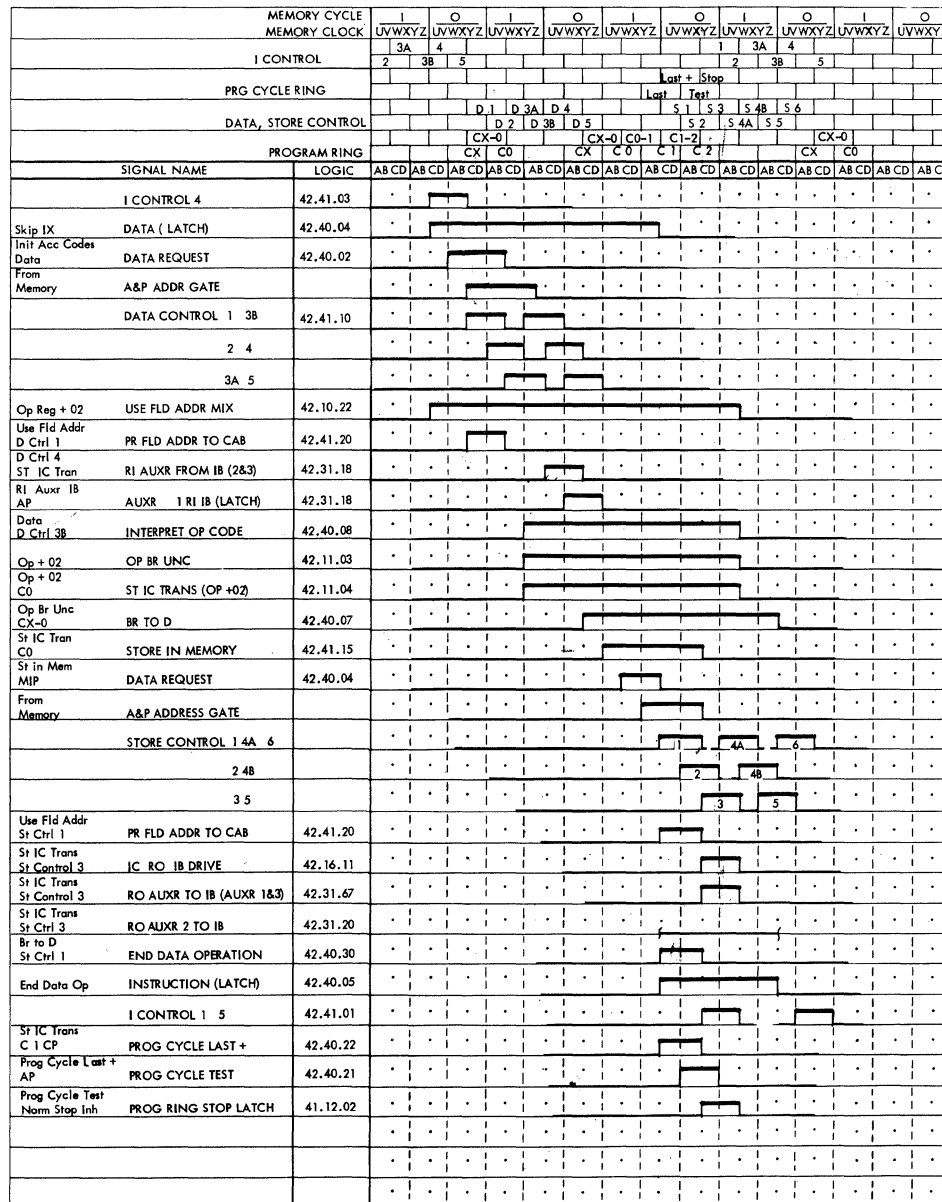
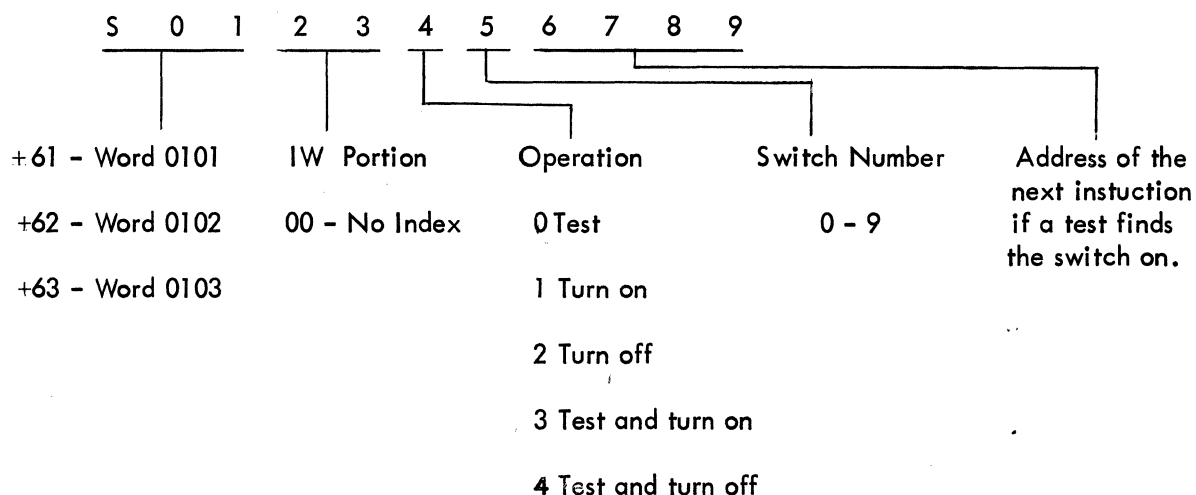


FIGURE 5.8-7. BRANCH AND LOAD LOCATION IN INDEX WORD

5.8.04 Electronic Switch Control, Op +61, +62, +63

There are 30 electronic switches available to the programmer. The on or off designation for each switch is stored in a storage word position. Words 0101, 0102 and 0103 are used for this purpose. Operation code +61 is used for testing and setting each of the ten switches in word 0101; +62 for word 0102; and +63 for word 0103. The instruction format is used by the electronic switch control in the following manner:



Note that this is a branch code only if position 4 contains a 0, 3 or 4. A zero in a control word position signifies switch off; a 1 (or 2.....9) signifies switch on.

Introduction

Electronic switch control is an initial access code, regardless of the digit in position 4 of the instruction. The control word is read out of core storage into the arithmetic register so that the operation specified by position 4 can be performed. A data-request signal is initiated which starts the data control ring in the usual way. The address sent to the core storage address trigger register must be generated in the address generator. If the Op Code is +61, the address 0101 is sent to core storage via the memory address bus. The selected control word reads into the arithmetic register during Data Control 4 time.

Once the control word is in the arithmetic register, the control digit specified by position 5 of the instruction word must be read out of the arithmetic register so that it can be tested. This is accomplished by setting the field ring to the same value as the units position of the field register. The output of the register is tested (if instruction word, position 4, is 0, 3 or 4) for zero or not zero. If the register output is zero (1 and 2 bits) a No Branch is signalled. If the output consists of 0, 3, or 6 bits, Branch to D is turned on. When position 4 of the instruction (tens position of the field register) is 1 or 2, no test is to be made so No Branch is automatically turned on without testing.

When the digit in position 4 of the instruction (tens position of field register) is 1, 2, 3 or 4, the selected digit in the arithmetic register must be reset to 0. If the digit in position 4 is a 1 or 3, the switch must be turned on following the reset. This is accomplished by inserting a 1 in the selected digit position.

On codes which involve the possibility of a change in the control word (position 4 is 1, 2, 3 or 4), the revised word must be stored back in the core storage location from which it was read out. This is done by initiating a store-in-memory operation. The storage address is generated from the Op code ; (+61) generates address 0101. The operation is completed by an arithmetic register read-out to the IB and the subsequent read-in to core storage.

<u>T. P. Value</u>	<u>Signal Developed</u>	<u>Operation Performed</u>
0, 3, 4	SWC Branch Codes	Test Arithmetic output for 0 - No Branch. , or Not 0 - Branch
1, 2, 3, 4	All SWC Sets	Reset selected arithmetic position. Gate 0 or 1 set switch input to selected arithmetic position.
1, 2	Set Control only	Turn on No Branch
1, 3	SWC Set on Control	Insert digit 1 on adder output.

Select and Test the Control Digit for 0, Not 0. After the control word is read into the arithmetic register from core storage via the IB, the program ring is started. Step C0 of the ring turns on field ring start. The output of field ring start switches with the units position output of the field register (Position 5 of the instruction) to set the field ring to the corresponding position. The serial output of the arithmetic register (Athr) is sampled by the selected field ring position to turn on the Athr output latches. The latch outputs are sampled at C1 time. Branch to D is turned on if 0, 3 or 6 bit latches are on. No Branch is turned on if 1 and 2 bit latches are on (digit 0).

Circuits for Electronic Switch Control

Figure 5.8-8 is a block diagram for the operation. The following switching operates in a manner similar to other initial access codes: data request, data control ring, program ring, interpret op code, program ring stop, and end data operation. A sequence chart of the operation is shown in Figure 5.8-9.

Read Out of the Control Word to the Arithmetic Register. This operation is conventional except for the generation of the control word address. A Generate Address 010X signal is sent to the 7602 where address 010X is generated and sent to the core storage address triggers via the memory address bus (Op code +61 causes Generate Address 0101, etc.).

Interpret the SWC Code. When Interpret Op Code is turned on at Data Control 3B time, an analysis is made of the tens position of the field register (Instruction Word Position 4). The following control signals are developed.

Reset and Gate RI of 0 or 1 to the Selected Position of the Control Word. If the tens position of the field register contains 1, 2, 3 or 4, the selected digit of the control word must be reset. Either a 1 or a 0 will then be read into the selected position from the adder output latches (forced on).

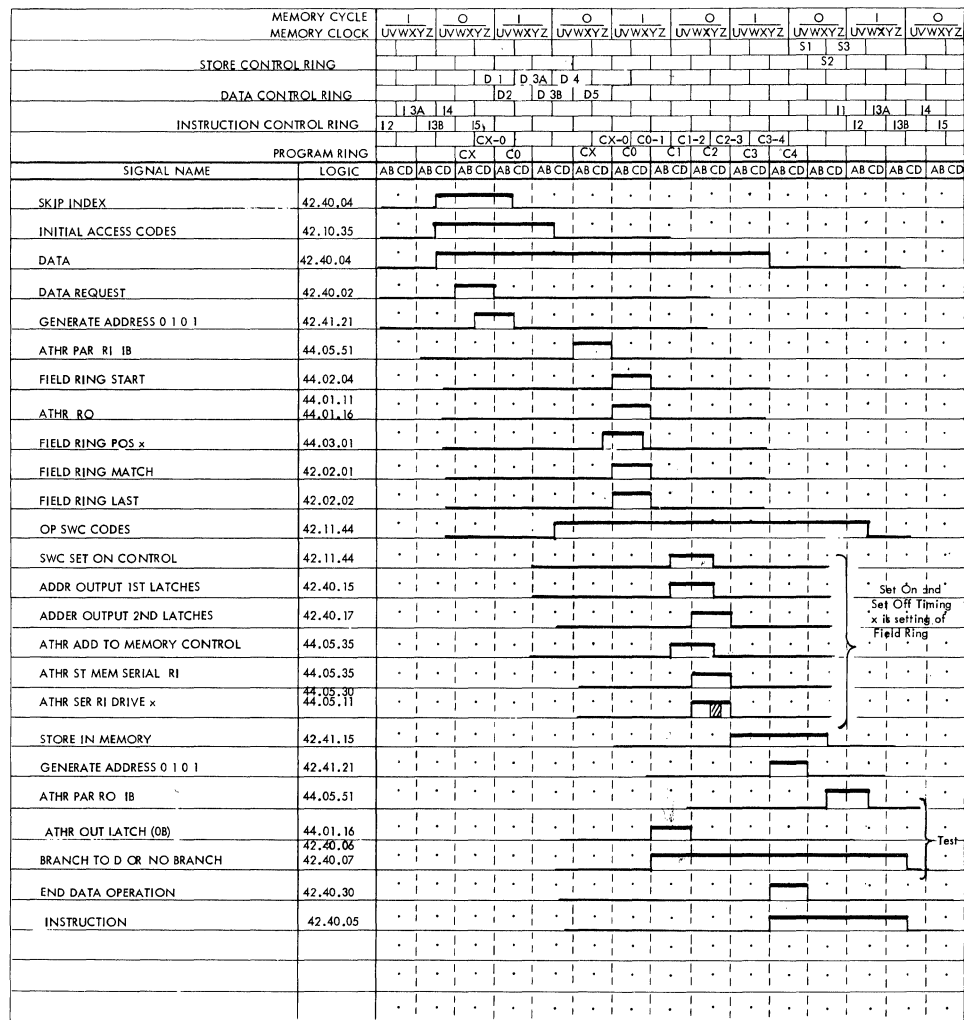


FIGURE 5.8-9. SEQUENCE FOR ELECTRONIC SWITCH CONTROL

All SWC Sets turn on Athr Add to Memory Control. This holds Athr Serial RO Drive off. The Serial RO capacitor is thus prevented from charging and the subsequent regen gate fails to regenerate the bit.

All SWC Sets also turn on Athr Store Memory Serial RI. This results in an Athr Serial RI Drive x, CP where x indicates Reg Position drivers. Read-in of 0 or 1 to the Athr from the adder output results.

Insertion of Digits 0 or 1 on the Adder Output. The 0 or 1 is inserted at the adder output 1st latches. When addition is not taking place, zeros are normally inserted on the adder output (1 and 2 bits). Thus, the normal zero insert circuit allows 0's to be read into the selected Athr position on Field Register TP 2 and 4 operations.

When the Field Register TP is 1 or 3, the SWC Set On Control forces the insertion of 0 and 1 bits at the adder output. SWC Set On Control blocks the insertion of the 2 bit, allows the normal insertion of the 1 bit, and forces the insertion of the 0 bit. The adder output data flow lines go to the Arithmetic Register In-Out Scanner where the 0 or 1 is inserted into the selected digit position of the control word.

Storage of the Revised Control Word in Core Storage. When the tens position of the field register is 1, 2, 3 or 4, the control word may have been changed by the performance of the instruction. Consequently, it must be stored back in the location designated by the Op Code; +61, word 0101, for example. This is accomplished by a normal store-in-memory operation with the storage address being generated as previously described.

5.8.05 Branch if Zero in Accumulator

Op +10, +20, +30 (BZ#)

The contents of the indicated accumulator is tested for zero. If the value is zero, the contents of program register is transferred to the instruction counter and the next instruction is taken from the I. C. If this value is not zero, the PR to IC transfer does not take place and the next instruction comes from the address in the IC. The sign of the accumulator is ignored when the zero - not zero test is made.

Introduction

The accumulator indicated by the tens position of the op code is tested for zero by reading out the accumulator to the significant digit scanner. If this scanner output is zero, the Branch to D latch is turned on, and the next instruction comes from the address in program register D.

Circuits for Branch Zero

Figure 5.8-10 block diagrams the circuits for the branch-if-zero operation. The operation is similar to other no-access branch codes in the manner in which Interpret Op Code, Program Ring Recycle, and Program Cycle Last are turned on. The transfer zero latch output causes the read-out of the selected accumulator to the arithmetic bus and the read-in to the significant digit scanner. If the accumulator contents are zero, the SDS Reg 0 signal turns on Branch to D. The operation of the significant digit scanner is described in the section on arithmetic codes. Figure 5.8-11 shows a sequence chart of the branch-if-zero operation.

5.8.06 Branch if Minus in Accumulator

Op - 10, - 20, - 30 (BM#)

The sign of the indicated accumulator is tested for plus, minus and alpha. If the sign is minus, the next instruction is taken from the address in PR D. If this sign is plus or alpha, the next instruction is taken from the address in the instruction counter.

Introduction

To accomplish the test for accumulator sign, the indicated accumulator sign cores are read out to one of the three Op'd Register Sign latches (alpha, plus, or minus). The latch output is then tested. Alpha or plus signs result in a No Branch signal; a minus output results in Branch to D.

Circuits for Branch if Minus

The operation of Ring Recycle, Interpret Op Code, Program Cycle Ring, and Op Branch Codes is the same as on other No-Access Branch Codes. Figure 5.8-12 summarizes the operation. The sequence of operation is shown in Figure 5.8-13.

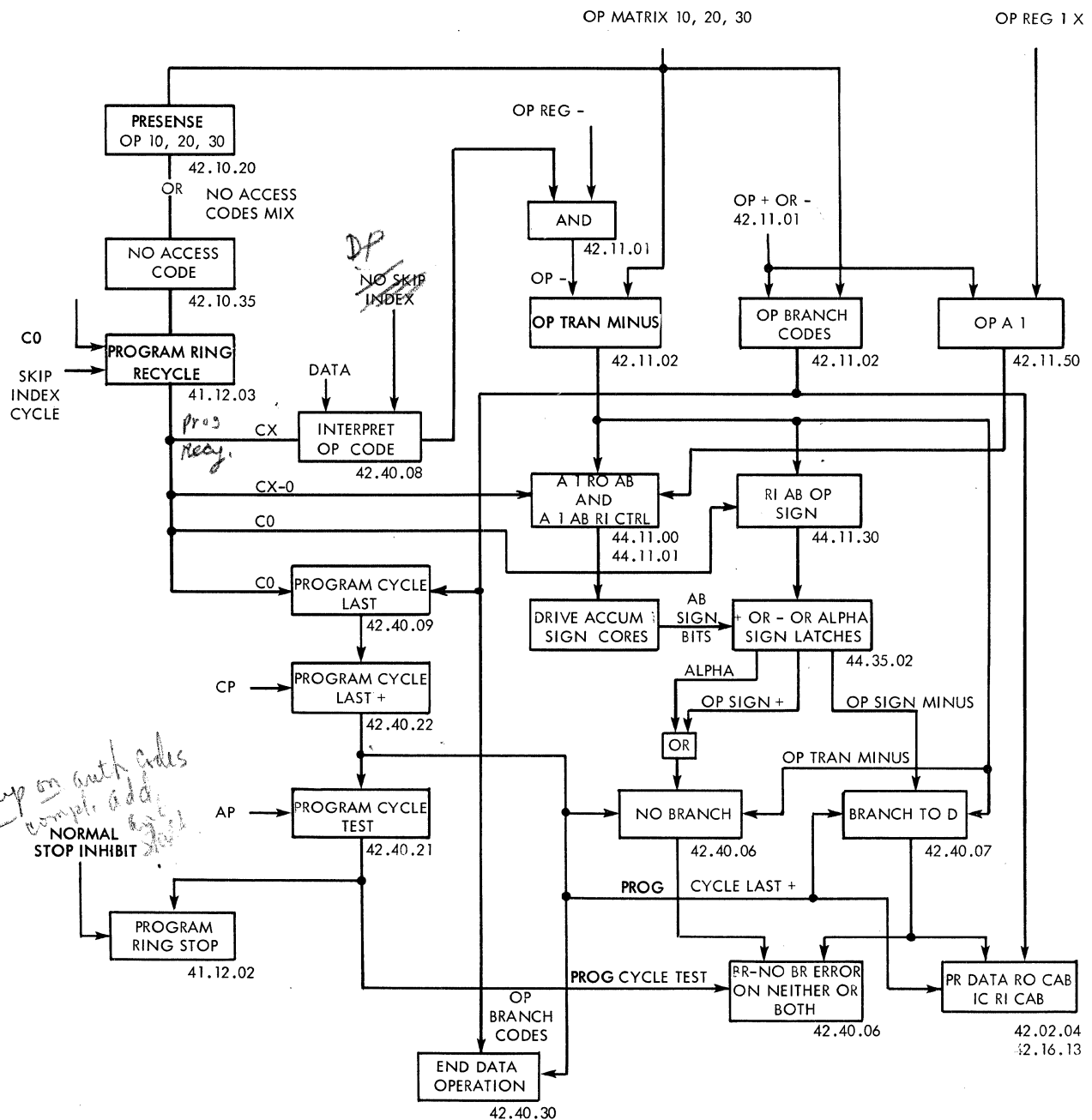


FIGURE 5.8-12. BRANCH IF MINUS IN ACCUMULATOR 1

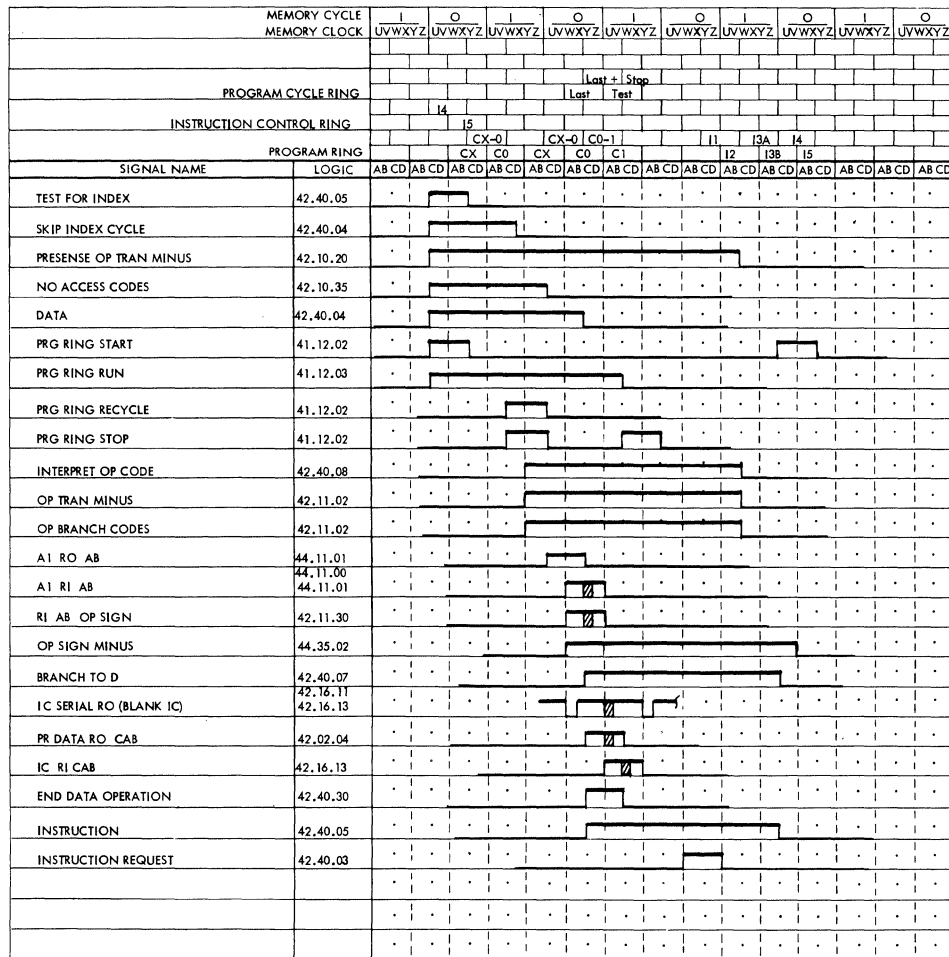


FIGURE 5.8-13. SEQUENCE FOR BRANCH MINUS

Seq Chart 3/1.00.49

5.8.07 Branch if Overflow in Accumulator

Whenever an accumulator overflows, the overflow indicator for that accumulator is automatically turned on. An accumulator overflow is developed if a true-add operation obtains an 11-digit result. The operation must be true add, not complement add.

Op +11, +21, +31 (BV#) Branch if Overflow in Accumulator #

This operation code tests the overflow indicator. If it is on, the next instruction is taken from the location specified by the data address portion of the instruction word; otherwise, the next instruction is taken from the address in the instruction counter.

Circuits for Branch if Accumulator Overflow

Figure 5.8-14 outlines the circuits to test the Accumulator 1 overflow indicator. This is a typical no-access branch code. The transfer overflow signal tests for A1 Ofl. with a Branch-to-D signal resulting if the indicator is on, and a No Branch if the indicator is off. A branch-overflow instruction turns off the indicator (if was on) after testing it.

The indicator is not reset by any of the tests.

Circuits for Branch High, Low, Equal

Each of these Op codes operate in the same manner as illustrated by Figure 5.8-15. For example, if the Op code is +40, Branch on Low, an Op Tr Low signal is developed which tests the low latch condition. If the latch is on, a Branch-to-D signal results. If the latch is off, a Hi-Lo-Eq No Branch Set signal results. Note that a compare code must have previously set any one of the latches in order to turn on No Branch Set.

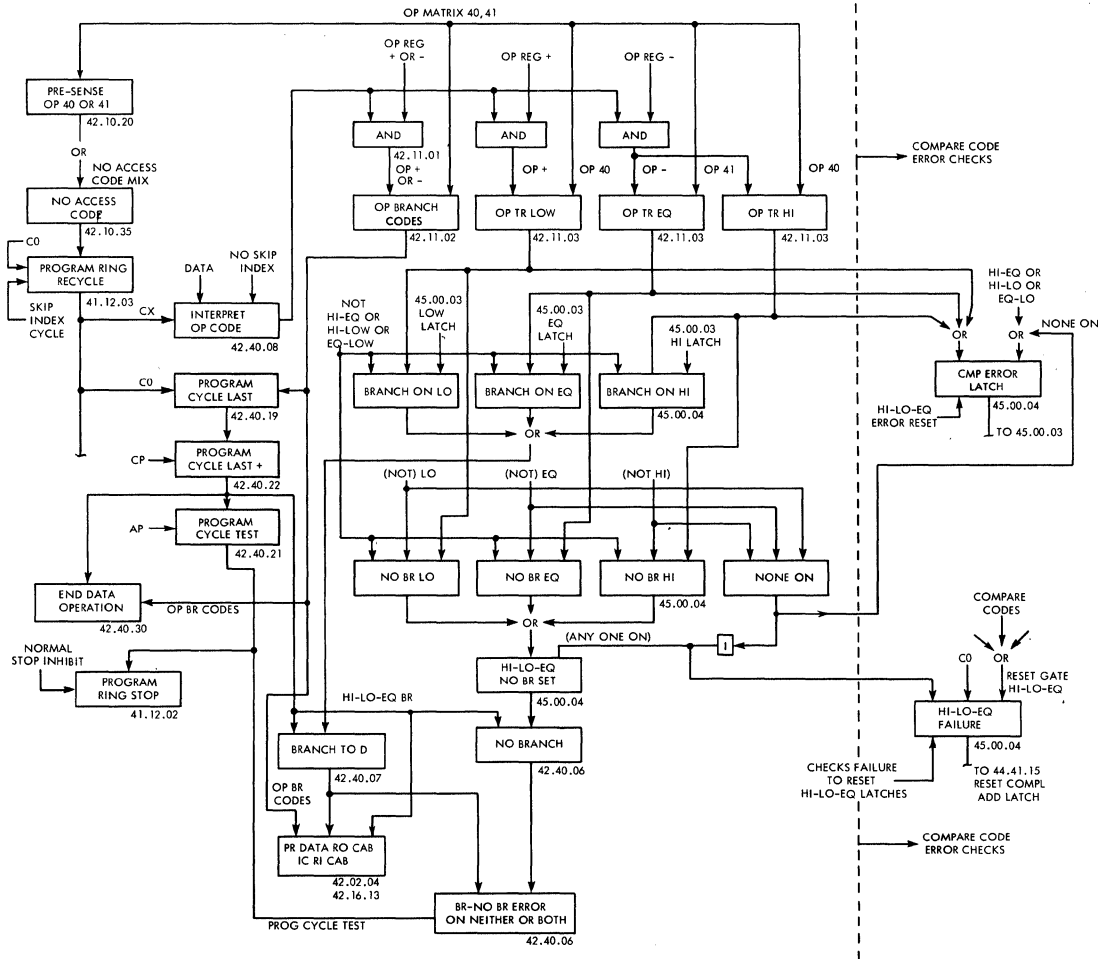


FIGURE 5.8-15. BRANCH ON HIGH, LOW, EQUAL

5.8.09 Field Overflow Control

Seq Chart 31.00.52

The field overflow latch is turned on when either of the following conditions occurs:

1. The number of significant digits in the accumulator is greater than field definition allows to be stored. This may happen on store or add-to-storage operations.
2. A carry is propagated beyond the high order position of the field defined in an add-to-storage operation.

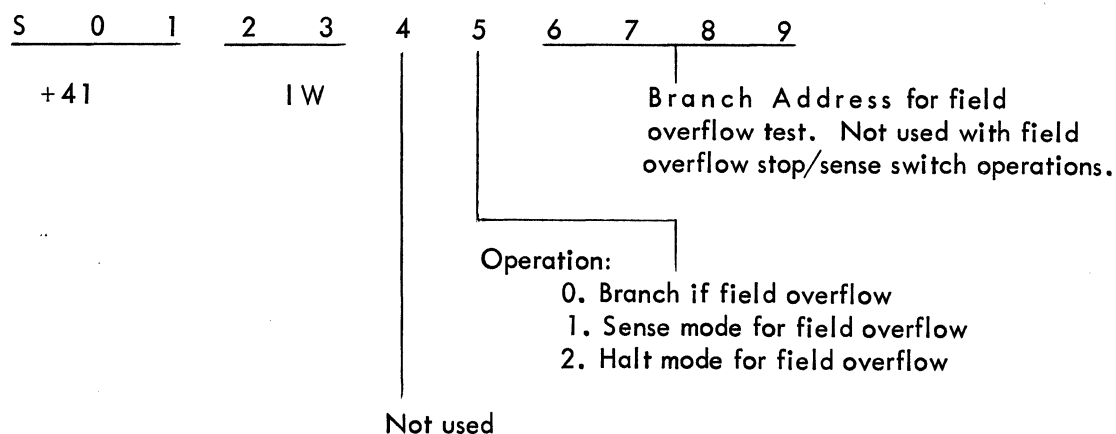
When either condition arises, only the digits that field definition allows are stored. The field overflow latch is turned on, and the machine either continues or stops, depending upon the setting of the field overflow stop/sense switch.

Op + 41 Field Overflow Control

This is an augmented code which serves the following two purposes:

1. Test the condition of the field overflow latch. If it is on, the next instruction is taken from the Address in the Program Register, otherwise from the Instruction Counter.
2. Set the field overflow stop/sense switch to Sense or to Stop.

The instruction format is utilized as follows:



A block diagram for field overflow control circuits is shown in Figure 5.8-16. This is a no-access code which is set up for operation in the usual manner. The Op + 41 signal switches with the position 5 output of the program register to provide one of the following series of operations:

0 in PR Position 5. The branch-on-field-overflow signal switches with the off condition of the field overflow latch to cause a No Branch signal. The branch or field overflow signal switches with the on condition of the field overflow latch to cause a Branch-to-D signal. If the field overflow indicator is on, the branch-on-field-overflow test will turn it off. Note that in both cases the overflow sense latch must be on to allow the program to continue.

1 in PR Position 5. The Set Sense on Field Overflow signal switches with a CX gate to turn the field overflow stop/sense switch to the On, or Sense position. As soon as the overflow sense latch goes to the sense position, the Sense output signal switches with the Set Sense On signal to turn on No Branch to allow the program to continue.

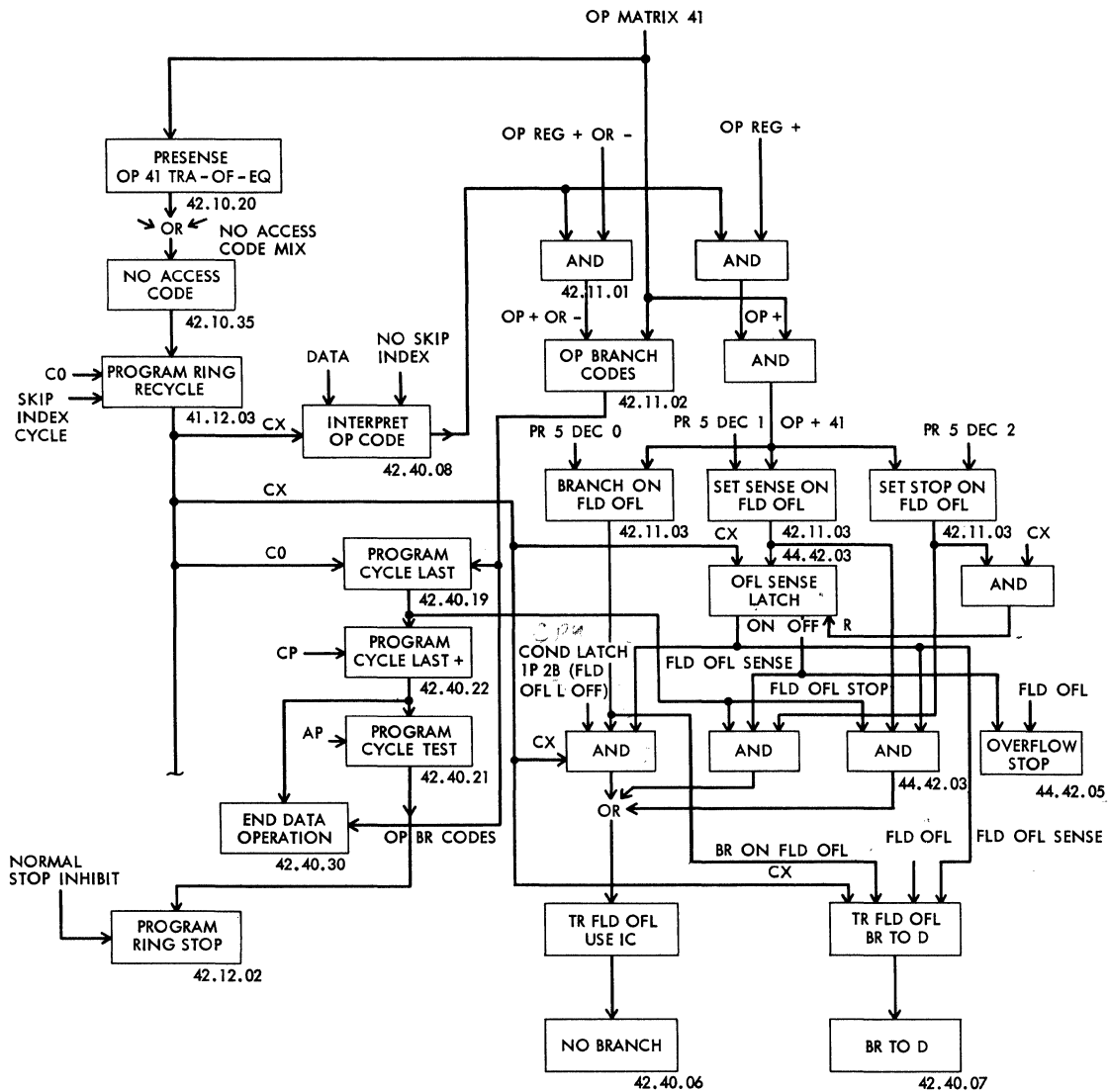


FIGURE 5.8-16. FIELD OVERFLOW CONTROL

Seq Chart 31.00.53

2 in PR Position 5. The Set Stop on Field Overflow signal switches with a CX gate to reset the Overflow Sense latch to its Off or Field Overflow Stop position. When the Overflow Sense latch goes to its Stop position, the field overflow stop signal switches with Set Stop on and Program Cycle Last to turn on No Branch. This allows the program to continue.

5.8.10 Branch on Alteration Switch or Channel Busy

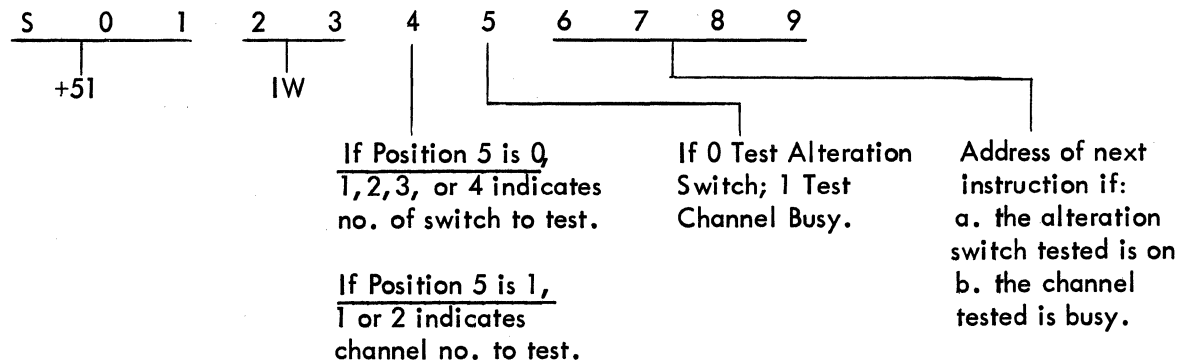
Op + 51 Pos 5 - 0 (BAS), Pos 5 - 1 (BCB)

The four alteration switches are located on the operator's console. They can be turned on or off manually.

The Channel Busy test interrogates one of the two tape/disk storage channels for a Busy condition.

Op + 51 is an augmented operation code which is further defined by positions 4 and 5 of the instruction as follows:

Positions of Instruction



Op + 51 Circuit Operation (Figure 5.8-17). The two variations of Op + 51 operate similarly to other no-access branch codes in regard to program ring recycle, interpret op code, and program cycle ring. The units and tens positions of the PR Field Register (Positions 4 and 5 of the instruction) are converted to decimal values. The Op + 51 ASCB signal is switched with the units position field register decimal outputs to create alteration switch test and channel test.

The tens position decimal 1 (or 2) output of the field register switches with a Sync 1 (or 2) Busy signal and, in turn, with the channel test signal to create a Branch-to-D signal. If neither channel is busy, a no branch is signalled.

The tens position decimal 1 (or 2, 3, or 4) output of the field register switches with switch 1 (or 2, 3, or 4) On and, in turn, with Alteration Switch Test to signal a Branch-to-D. If all four alteration switches are off, No Branch is turned on.

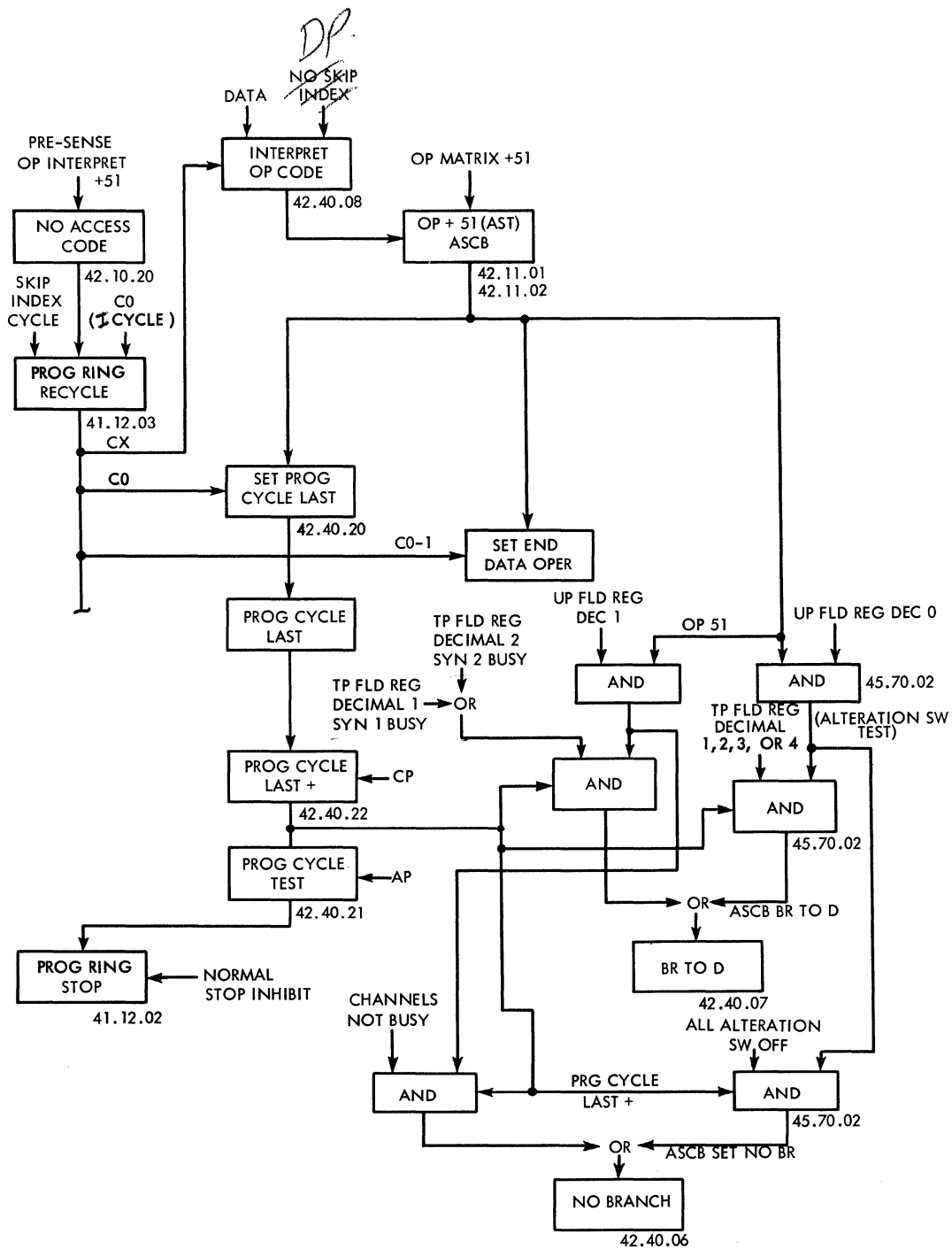
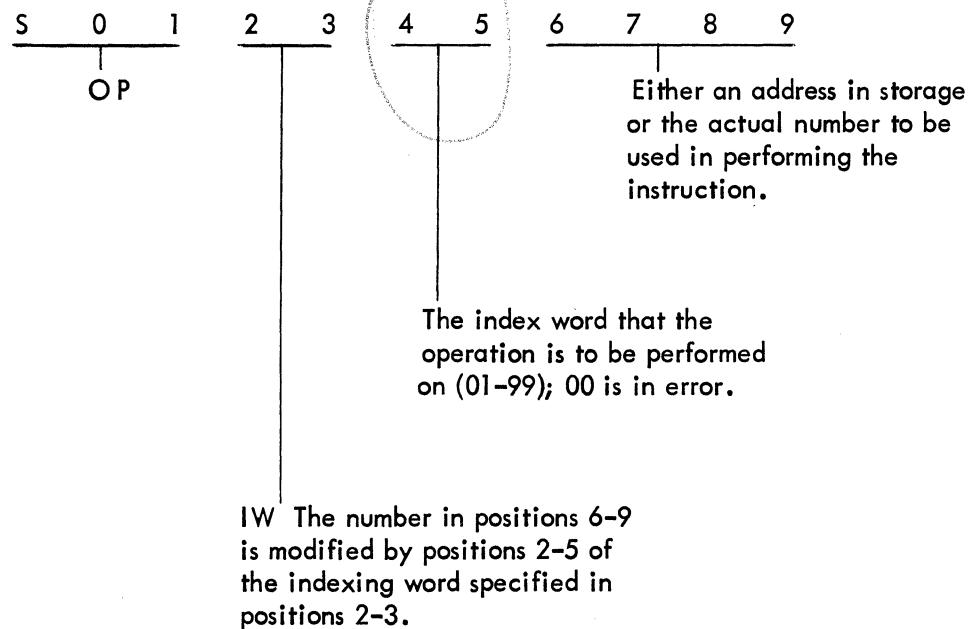


FIGURE 5.8-17. ALTERATION SWITCH, CHANNEL BUSY, BRANCH

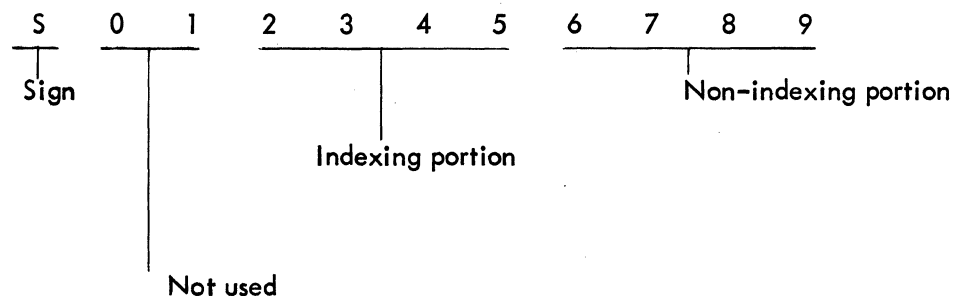
Seq Chart 31.00.54

5.9.00 INDEXING INSTRUCTIONS

The index word operation codes are used to load, unload, modify, test, etc., any of the 99 index words in storage. Positions 4-5 of the index instruction word are used to designate which of the 99 words is to be operated upon. Note that positions 6-9 of an index instruction can be indexed by the indexing word specified by the IW portion (positions 2-3) of the instruction in the same manner as any other operation code. Following is the format of an indexing instruction word.



The format of the index word specified by positions 2-3 (IW) or 4-5 of an indexing instruction is as follows:



The indexing portion is used to modify positions 6-9 of the instruction being indexed. The non-indexing portion is used for decrements, limits and constants. The sign of the index word is controlled by the operations on the indexing portion. Any operation on the indexing portion which results in a sign change changes the sign of the entire word.

5.9.01 Branch Compared Index Word

Op - 43 (BCX) Branch Compared Index Word

The value of the indexing portion of the index word specified by positions 4-5 of the instruction is compared to the value of the non-indexing portion. If the indexing portion is less than, or equal to, the non-indexing portion, the next instruction is taken from the address in program register D. If the indexing portion is greater, the next instruction is taken from the instruction counter.

Introduction

Op Code - 43 is an initial access code, requiring the index word specified by positions 4-5 of the instruction be brought from core storage to the auxiliary register. A complement add operation is then started to compare the indexing portion to the non-indexing portion. Auxiliary 2 (indexing portion) is read out serially to Channel 2 and complemented at Adder Entry B. At the same time, Auxiliary 3 (non-indexing portion) is read out serially to Channel 1 and to the Adder Entry A. The following arithmetic operation is carried out at the adder when the indexing portion is less than the non-indexing portion:

Auxr	Adder	
Contents	Operation	
0348	0348	Auxiliary 3, non-indexing, Channel 1
0347	9653	Auxiliary 2, indexing Channel 2
	$\leftarrow C$ 0001	

Branch if digit 2-5 = or < than D6-9

As long as the indexing portion remains less than, or equal to, the non-indexing portion, a carry from the adder causes a Branch-to-D signal and the ensuing use of the D-address. When the indexing portion becomes greater than the non-indexing portion, the no-carry adder output causes a no-branch signal and the I. C. address is used.

Circuits for Branch Compared Index Word

Compare 2-5 against 6-9

Figure 5.9-1 shows a block diagram of the circuits used for this operation. The sequence of operation is shown in Figure 5.9-2. The following objectives must be accomplished:

Read Out Index Word from Core Storage. The index word designated by the field address portion of the -43 Op code instruction is read out to the auxiliary register by a conventional memory access operation as shown in Figure 5.9-1.

Interpret the Op Code. The D Control 3B output of the data ring turns on Interpret Op Code in the normal manner. The Interpret Op Code signal then switches with the output of the op code register to develop a Tran IP less or Equal FP signal. Similarly, Op IX Codes and Op Branch Codes signals are brought up. Together, these three lines control the entire operation.

Complement Auxiliary 2. The complement-add latch is turned on by Tran IP Less or Eq FP. The output of this latch controls the complement add operation as described in section 5.1.01.

		MEMORY CYCLE MEMORY CLOCK																										
		I			O			I			O			I			O			I			O					
		UVWXYZ			UVWXYZ			UVWXYZ			UVWXYZ			UVWXYZ			UVWXYZ			UVWXYZ			UVWXYZ					
		13A 14 15															11 13A 14			12 13B 15								
INSTRUCTION CONTROL RING																												
					D1 D3A D4																							
DATA CONTROL RING					D2 D3B D5																							
PROGRAM CYCLE RING																	Last + Stop											
					X-0			X-0			0-1			1-2			2-3			3-4			4-5			Last Test Stop +		
PROGRAM RING																												
SIGNAL NAME		LOGIC		AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	AB	CD	
PRE-SENSE TRAN IX CODES		42.10.24																										
USE FIELD ADDRESS		42.10.35																										
DATA		42.40.04																										
DATA REQUEST		42.40.04																										
FIELD ADDRESS TO CAB		42.41.20																										
RI AUXR FROM IB		42.31.19 42.31.18																										
COMP ADD		44.41.15																										
ADVANCE AUXR 2		42.31.11 42.31.51																										
ADVANCE AUXR 3		42.31.12 42.31.51																										
AUXR 2 REGEN RI 2		42.31.55 42.31.26																										
AUXR 3 REGEN RI		42.31.55 42.31.29																										
CARRY 1 ST (SAMPLED BY C3-4)		42.40.13 44.41.01																										
BRANCH TO D		42.40.13 42.40.07																										
DATA ADDRESS TO CAB		42.41.20																										
IC RI CAB		42.16.13																										
END DATA OPERATION		42.40.30																										
INSTRUCTION		42.40.05																										
NO BRANCH (IF NO CARRY)		42.40.13 42.40.06																										
AUXR 2 RO CH 2		42.31.53 42.31.24																										
AUXR 3 RO CH 1		42.31.54 42.31.13																										

FIGURE 5.9-2. BRANCH COMPARED INDEX WORD, OP - 43

5.9.02 Branch if Indexing Portion is Non-Zero

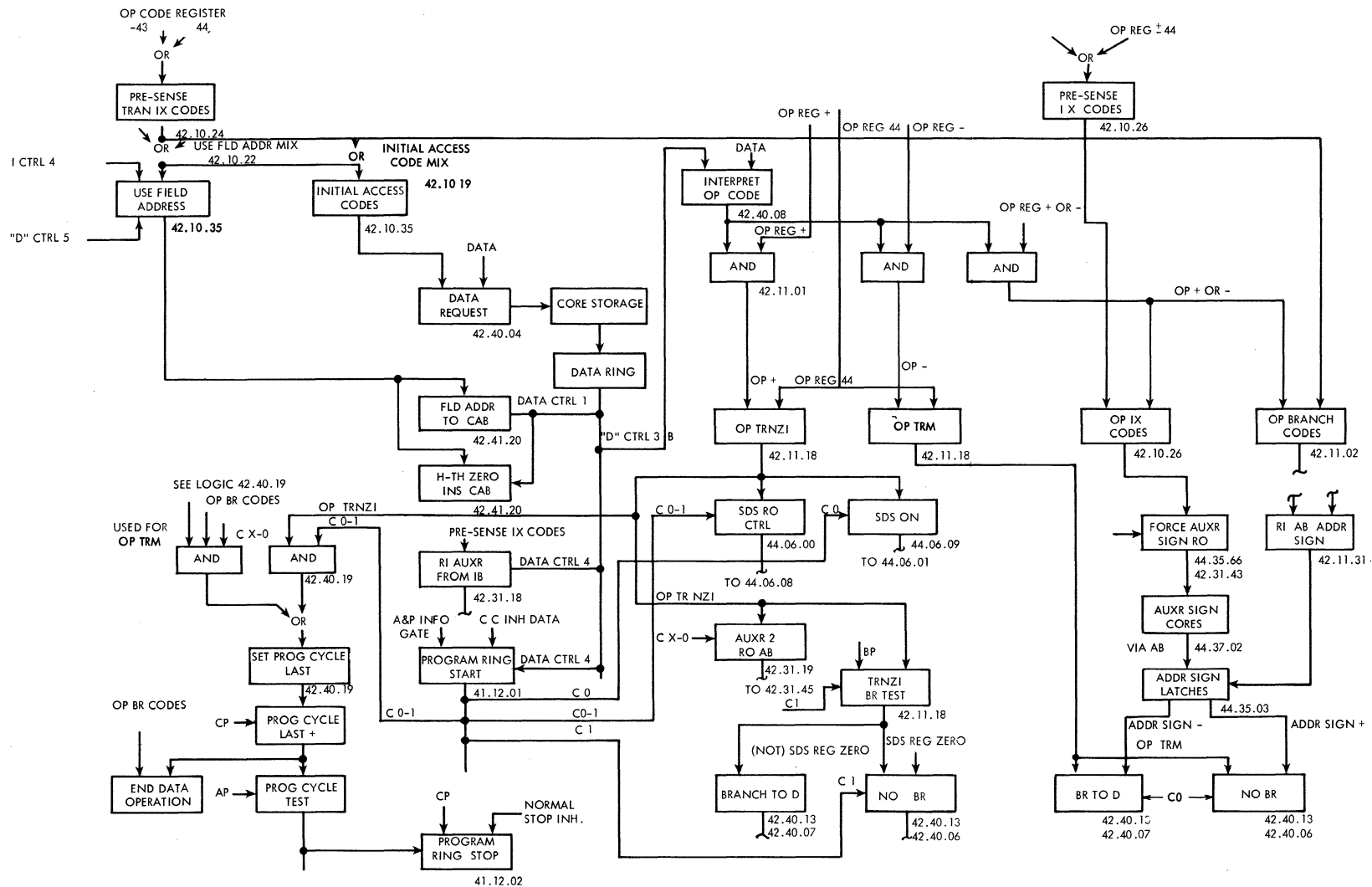
Op +44 Branch if Index Word Indexing Portion is Non-Zero

The indexing portion of the index word specified by the field register is tested for zero. If it is not zero, a Branch-to-D occurs and the next instruction is taken from the address in program register D; if it is zero, the next instruction is taken from the instruction counter.

Introduction

Op +44 is an initial access code, requiring that the index word specified by the field address portion of the instruction be read out to the auxiliary register via the IB. The index word is next read out of the auxiliary register to the arithmetic bus and is scanned by the significant digit scanner to determine whether it is zero or non-zero. A Branch-to-D results if the SDS Register is non-zero.

FIGURE 5.9-3. BRANCH IF INDEX WORD INDEXING PORTION IS
NON-ZERO (+44 TRNZI)
BRANCH IF INDEX WORD IS MINUS (-44 TRM)



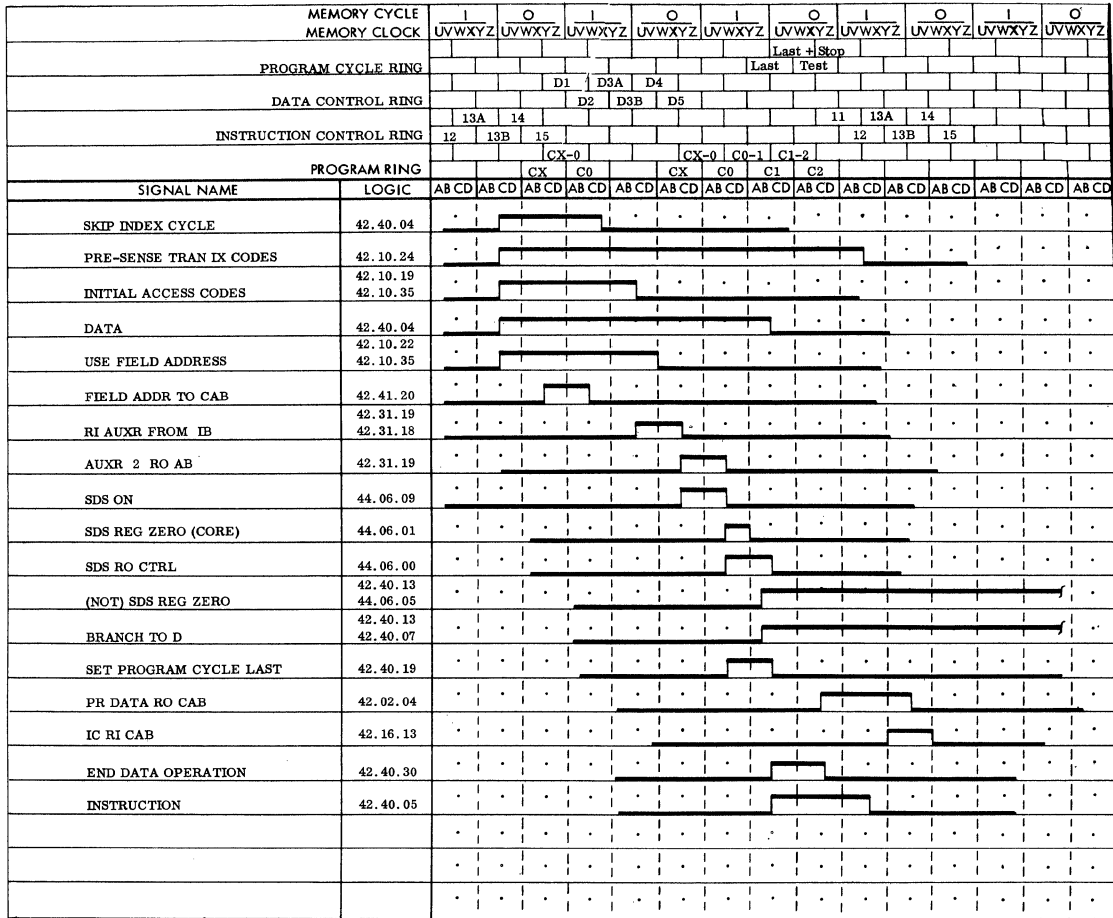


FIGURE 5.9-4. SEQUENCE FOR BRANCH IF INDEX WORD IS NON-ZERO, OP + 44

Circuits for Branch if Indexing Portion is Non-Zero

A block diagram of the operation is shown in Figure 5.9-3. Note that Branch if Index Word is Minus is shown on the same figure. See Figure 5.9-4 for the sequence of operation.

The designated index word is read out of core storage to the auxiliary register by the usual data request operation as illustrated in Figure 5.9-3. The 3B step of the data control ring turns on Interpret Op Code. The Op Register outputs then switch with Interpret Op Code to turn on the Op TRNZI signal which, along with Op IX Codes and Op Branch Codes, controls the entire operation. The following objectives are accomplished:

Read Out Auxr 2 to AB. This is done at CX-0 time.

Control the Significant Digit Scanner. The scan for zeros takes place at C 0 time. The operation of the SDS unit is discussed in section 2.0.00.

Test the SDS Output. As shown in Figure 5.9-3, the Branch-to-D signal results if the SDS Register is not zero and a No Branch if the SDS Register is zero. Because the zero latch may not be set at AP time, the test for Branch is at BP time to allow the latch to establish a Zero condition.

Introduction and Circuits(Figures 5.9-6 and 5.9-7)

Op + 45 is an initial access code, involving the use of the data address and a normal memory request operation to transfer the index word to be loaded from core storage, or any of the three accumulators, to the auxiliary register. The Data Control Ring, step 3B, turns on Interpret Op Code which then switches with the Op Code Register output to cause an Op LIW or UIW signal. This initiates a store-in-memory operation, using the field address to store the contents of the auxiliary register in core storage.

5.9.05 Index Word Unload

Op - 45 Index Word Unload (XU)

The contents of the core storage or accumulator location specified by the address in program register D are replaced by the contents of the designated index word.

Introduction and Circuits(Figures 5.9-8 and 5.9-9)

Op - 45 is an initial access code, involving the use of the field address and a memory request operation to transfer the index word being unloaded from the index word location to the auxiliary register. The normal Interpret Op Code sequence results in the development of the OP LIW or UIW signal as shown in Figure 5.9-8. A store-in-memory operation follows in which the index word being unloaded is transferred from the auxiliary register to the core storage location designated by the data address.

If the D address is an accumulator address, A-P INH Access is brought up by switching Store 1 and Addr 999X and the auxiliary register is transferred via the arithmetic bus to the accumulator address in the CAB latch register.

5.9.06 Index Word Zero and Add To-Subtract From Indexing Portion

Op + 46 Index Word Zero and Add to Indexing Portion (XZA)

The number in the data address positions of the program register (6 - 9) replace the previous contents of the indexing portion of the designated index word. The sign of the index word is set to plus and the remaining positions are unchanged.

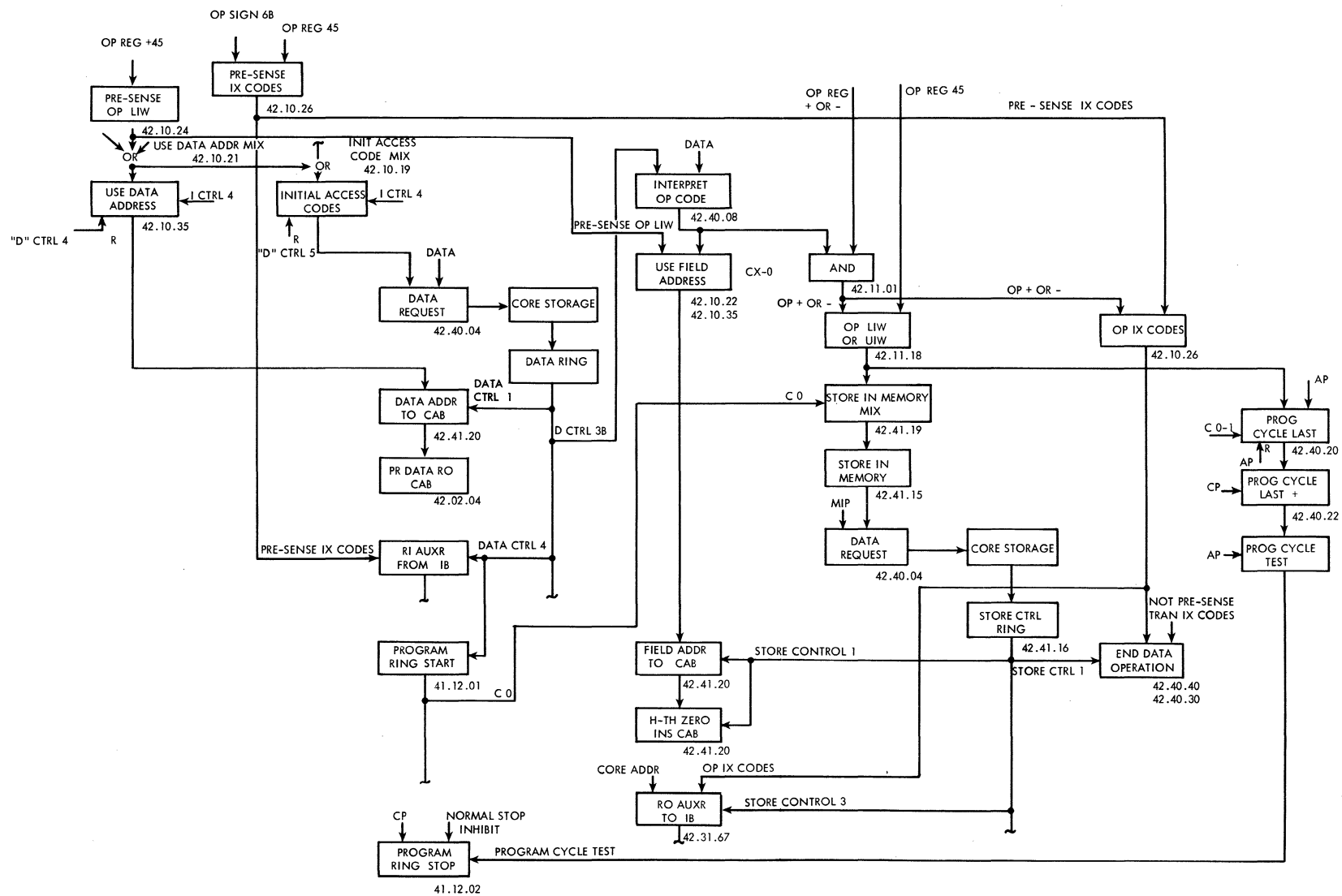
Op - 46 Index Word Zero and Subtract from Indexing Portion (XZS)

This code operates the same as + 46 except that the sign of the index word is set to minus.

Introduction

Op $\pm$ 46 is an initial access code, requiring that the index word designated by the field address be transferred to the Auxiliary Register by a memory access operation. The contents of program register D are then added to zeros in the adder and the adder output is brought back to Auxiliary 2. In the process, the previous contents of Auxiliary 2 are reset. The sign of the auxiliary register is set to + (if code + 46) or - (if code - 46). The auxiliary register is then stored back in the core storage address designated by the field address.

FIGURE 5.9-6. INDEX WORD LOAD, OP + 45



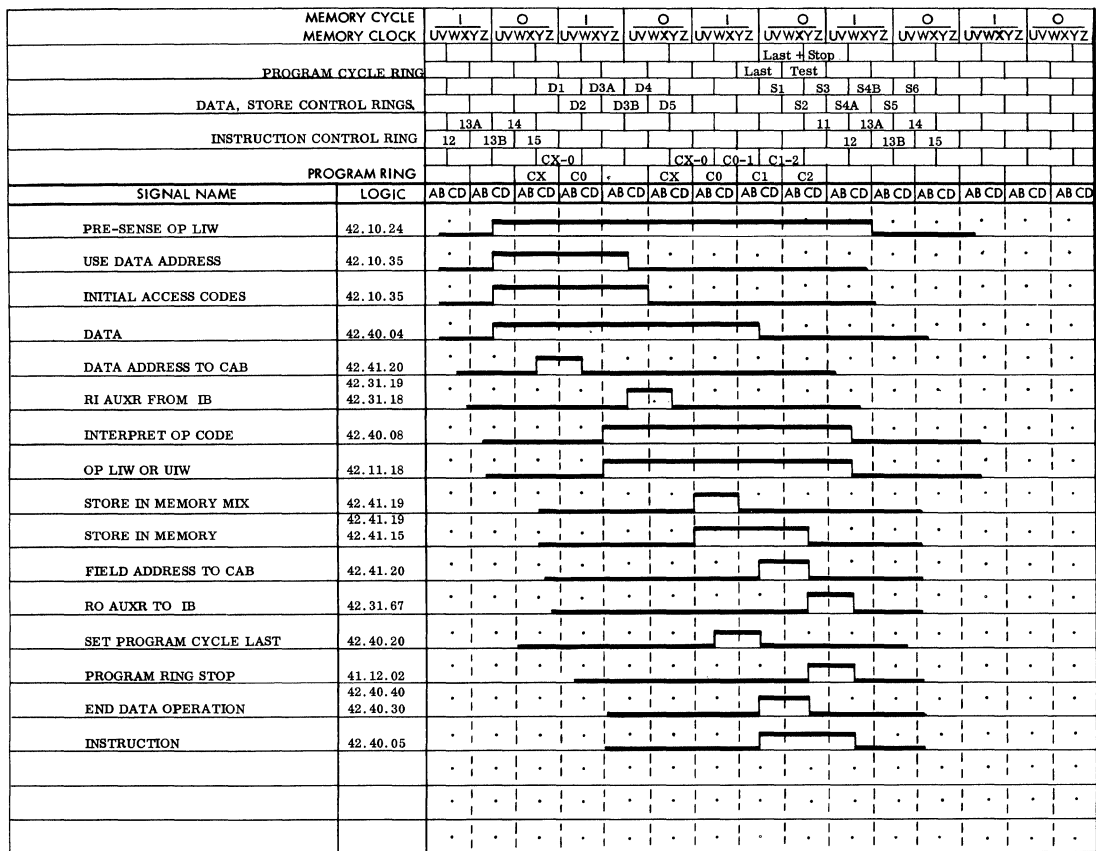


FIGURE 5.9-7. SEQUENCE FOR INDEX WORD LOAD, OP + 45

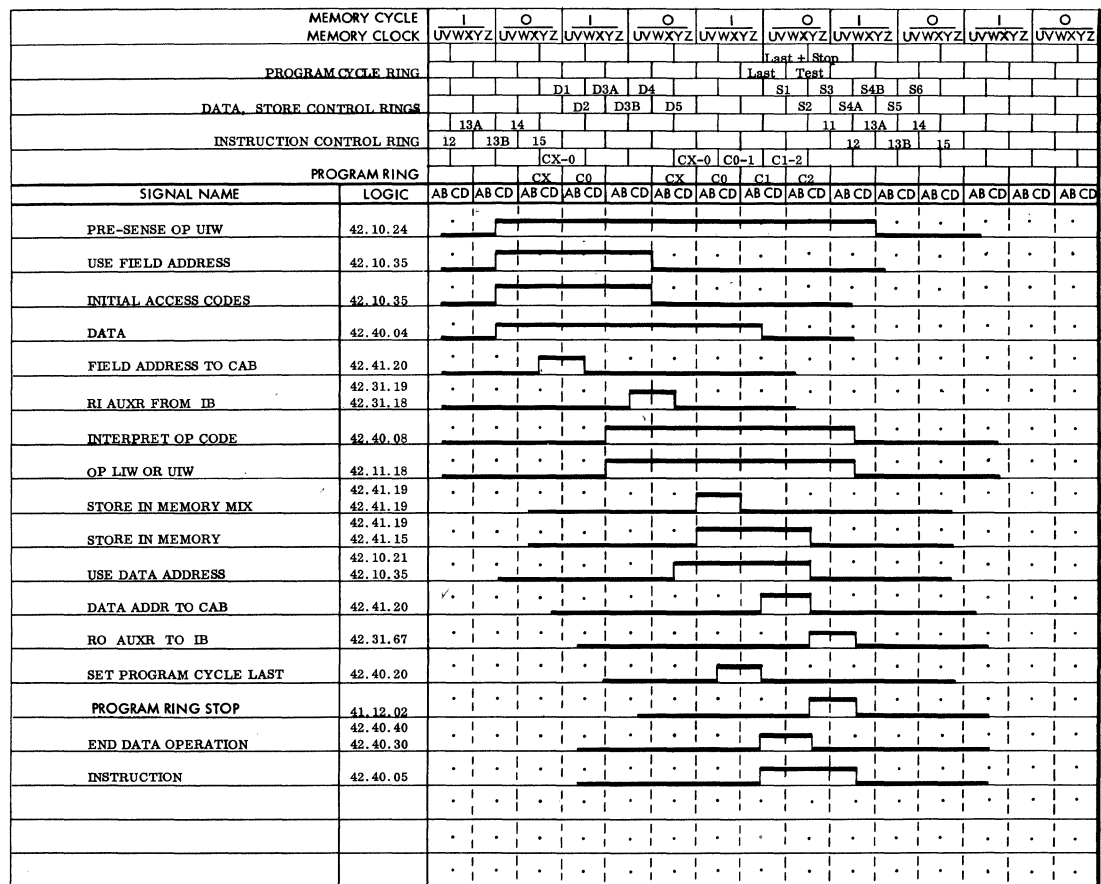
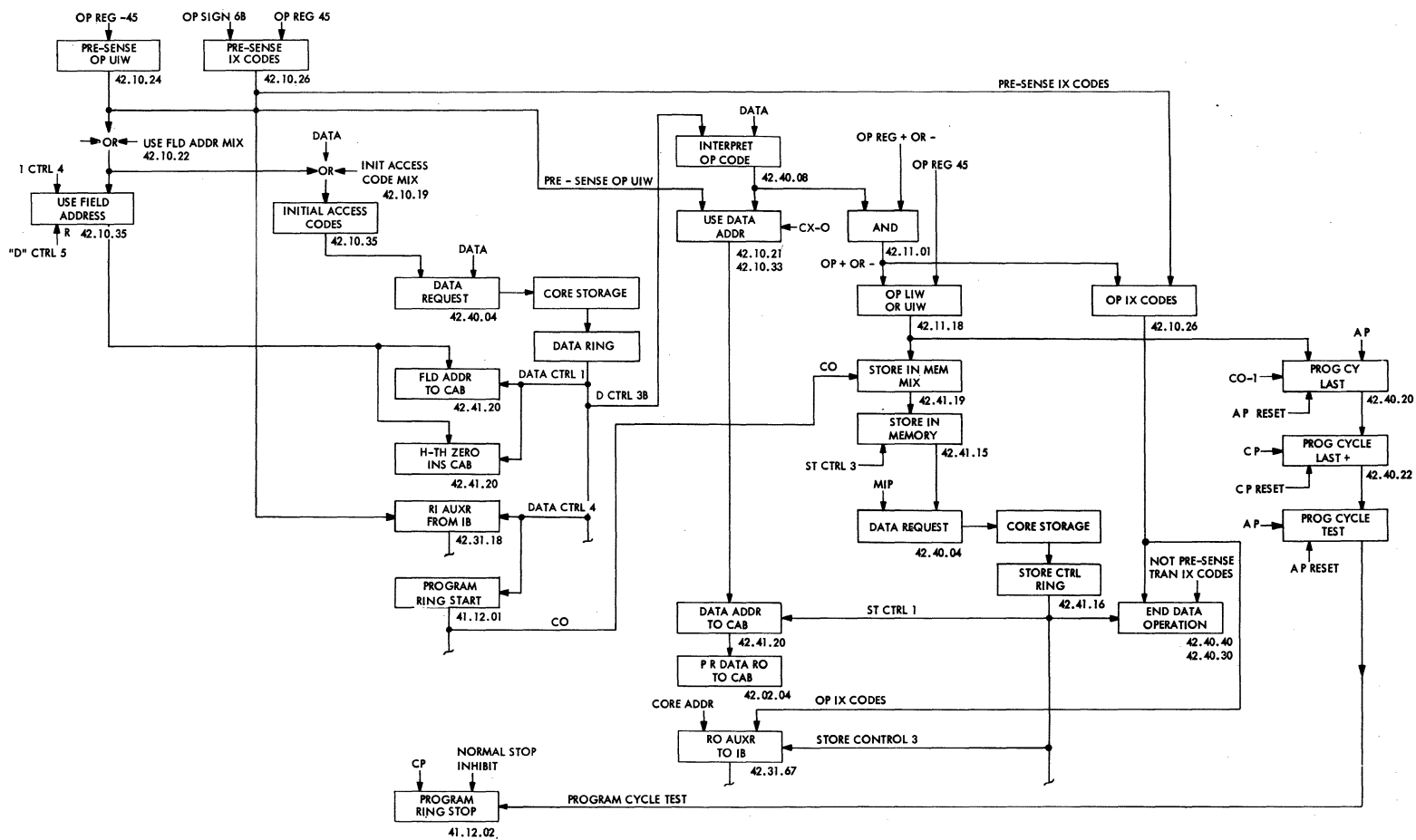


FIGURE 5.9-8. SEQUENCE FOR INDEX WORD UNLOAD, OP -45

FIGURE 5.9-9. INDEX WORD UNLOAD, OP-45



Circuits for Op Codes ± 46 , RADI and RSDI

A block diagram and sequence chart of the operation are shown in Figures 5.9-10, 11. After the memory access operation has transferred the designated index word to the auxiliary register, the following objectives must be accomplished:

Interpret the Op Code. The Data Control Ring 3B output times the turn on of the Interpret Op Code signal. The outputs of the Op Code Register are then switched with Interpret Op Code to develop the following operation control signals:

1. Op RADI or Op RSDI
2. Op RADI (If Op code + 46)
3. Op RSDI (If Op code - 46)
4. Op IX Codes

Set up Adder for True Add. The True Add latch is turned on at C0 time. This permits the contents of program register D to read into the adder, from Channel 2, in true form.

Read Out PR Data to Channel 2. This is accomplished by the following signals (Figure 5.9-10):

1. PR Data Serial Advance causes the PR contents to shift serially to the adder B input, units position first.
2. PR Data Serial Regen causes the units position data to be regenerated to the high order end of the PR data register.
3. RO PR Data to Channel 2 gates the PR Data to Channel 2 and thence to the Adder B entry.

Reset Auxr 2 and Insert Zeros on Channel 1. Auxr 2 is advanced but is not gated to Channel 1; instead, zeros are inserted by the Ch 1 Ins 0 circuits to form the Adder A entry.

Gate Adder Output to Auxr 2. The Auxr 2 RI Adder gate, turned on at C0-1 time, gates the Adder output back to Auxr 2.

Set the Auxr Sign. If the Op RADI signal is on, a plus sign is inserted in the Auxr sign cores. If the Op RSDI signal is on, a minus sign is inserted.

Store the Index Word Back in Memory. This is a conventional store-in-memory operation, using the field address to store the revised index word back in memory. The operation is started at C3 time by the IX Codes store-in-memory signal.

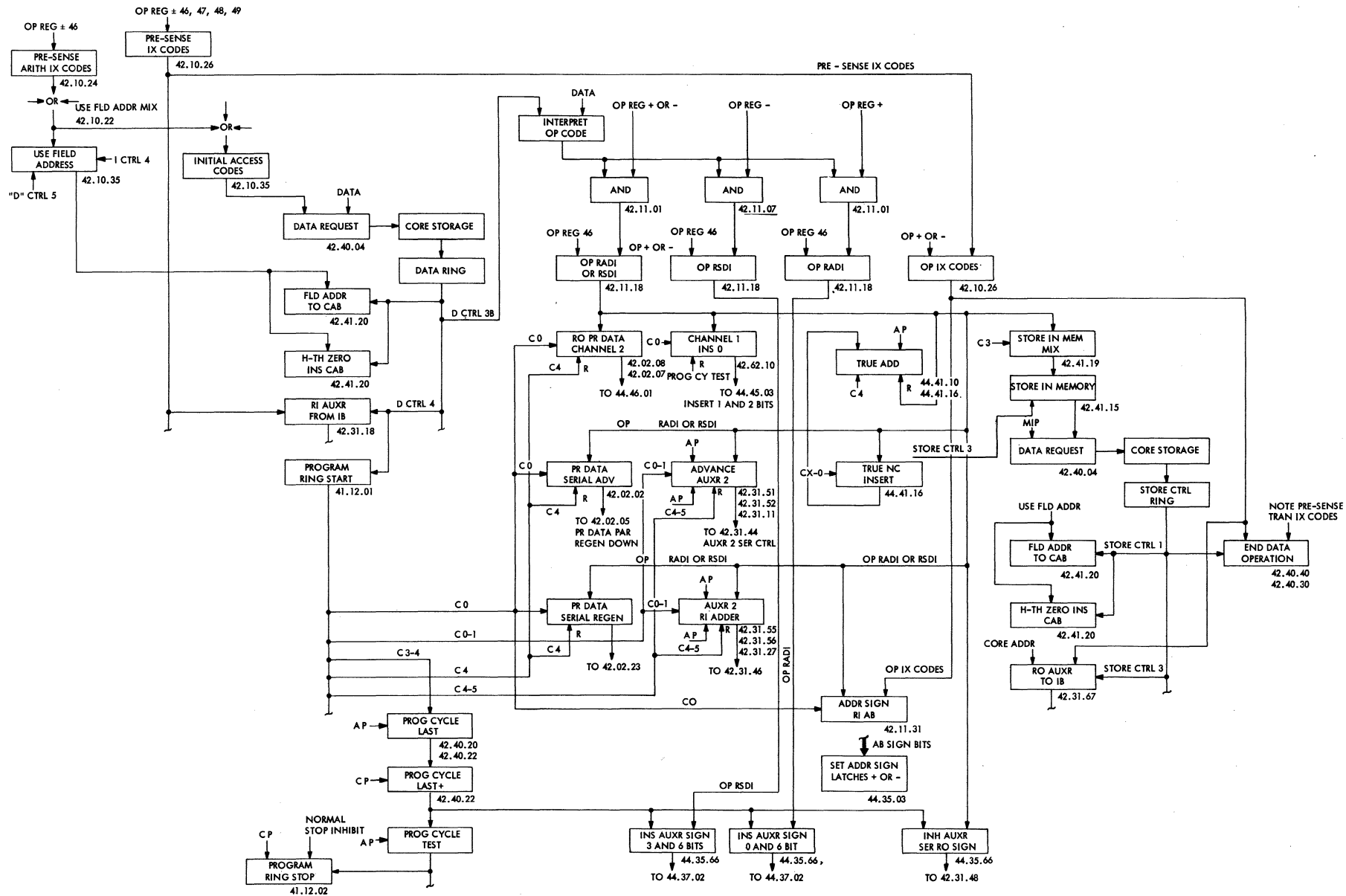
5.9.07 Index Word Add To - Subtract From Indexing Portions

Op + 47 Index Word Add to Indexing Portion (XA)

The contents of Program Register D is algebraically added to the indexing portion of the designated index word.

1. The sign of the contents of program register D is considered plus.
2. The sign of the designated index word is considered.
3. The sign of the result becomes the sign of the index word.

FIGURE 5.9-10. INDEX WORD ZERO AND ADD TO - SUBTRACT
FROM INDEXING PORTION, OP $\pm$ 46



[illegible]

FIGURE 5.9-11. SEQUENCE FOR INDEX WORD ZERO AND ADD TO -
SUBTRACT FROM INDEXING PORTION, OP $\pm$ 46

The result of the add operation replaces the previous indexing portion of the indexing word and the revised index word is stored back in core storage.

Op - 47 Index Word Subtract from Indexing Portion (XS)

Op - 47 is an initial access code in which the index word designated by the field address portion of the program register is read out from core storage to the auxiliary register. The signs of the factors are analyzed and either the true or complement-add latch is turned on. If a complement-add operation takes place, a recomplement cycle may be necessary. No-carry from the high order on a complement add operation signals the beginning of the recomplement cycle.

Data flow to the adder is accomplished by serially advancing program register D to the Adder B Entry via Channel 2 at the same time that Auxiliary 2 is being serially advanced to the Adder via Channel 1 to the Adder A Entry. The program register D digits are regenerated and the Auxiliary 2 digits are replaced by the result from the Adder output. The result in the auxiliary register is stored back in the core storage location specified by the field address.

Circuits for Op Codes $\pm$ 47 ADI and SDI

Figures 5.9-12, 13 and 14 outline the circuits and sequence of operation for Add to and Subtract from indexing portion. The following major objectives must be accomplished:

Interpret the Op Code. The outputs of the Op Code Register are switched with Interpret Op Code to develop the following operation control signals.

1. Op ADI or SDI
2. Op ADI (if Op Code + 47)
3. Op IX Codes

Turn on Either True Add or Complement Add. The Op Register sign together with the signs of both factors are analyzed to determine whether the true-add or complement-add latch should be turned on. Refer to the section on Sign Control in 5.1.00.

Control Data Flow of AUXR to the Adder. This is accomplished as shown on Figure 5.9-12 by development of the following gates:

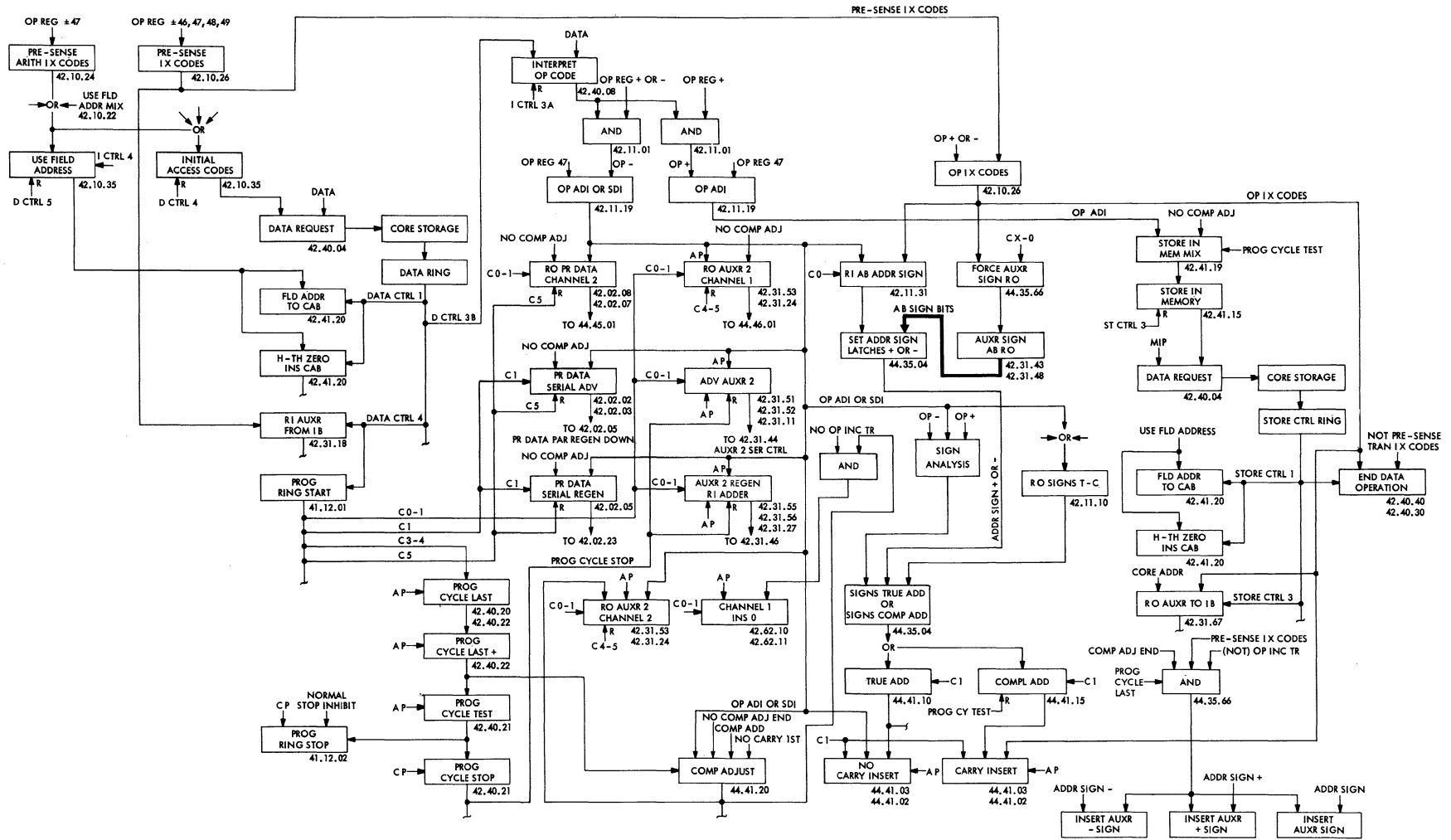
1. Advance Auxiliary 2
2. RO Auxr 2 Channel 1
3. True-Add or Complement-Add

Control Data Flow of PR Data to the Adder. Figure 5.9-12 shows the control of PR data to the adder. The following gates are needed:

1. PR Data Serial Advance
2. RO PR Data Channel 2
3. PR Data Serial Regen

Gate Adder Output Back to Auxr 2. This is accomplished by the Auxr 2 RI Adder signal.

FIGURE 5.9-12. INDEX WORD ADD TO - SUBTRACT FROM
INDEXING PORTION, OP $\pm$ 47



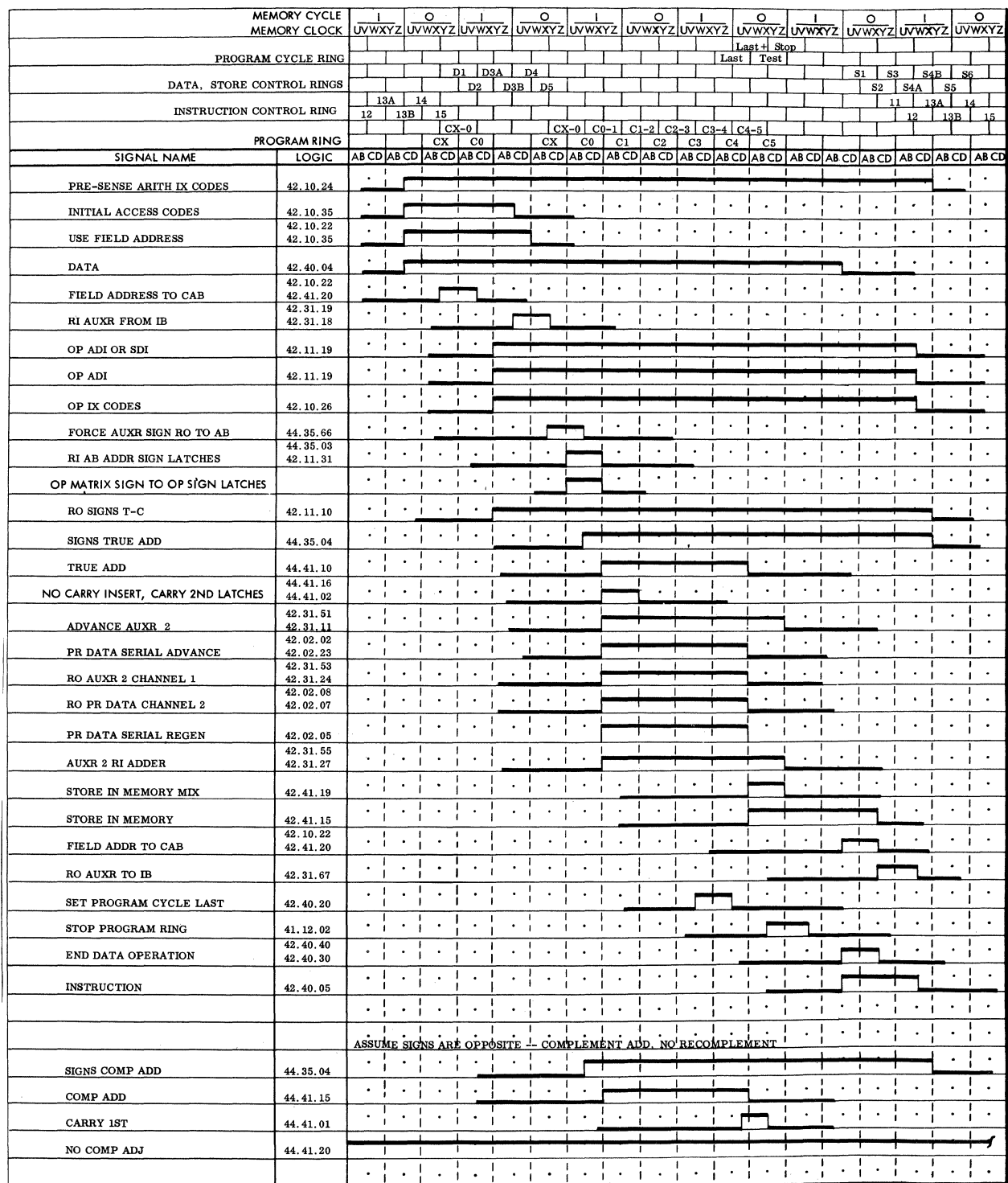
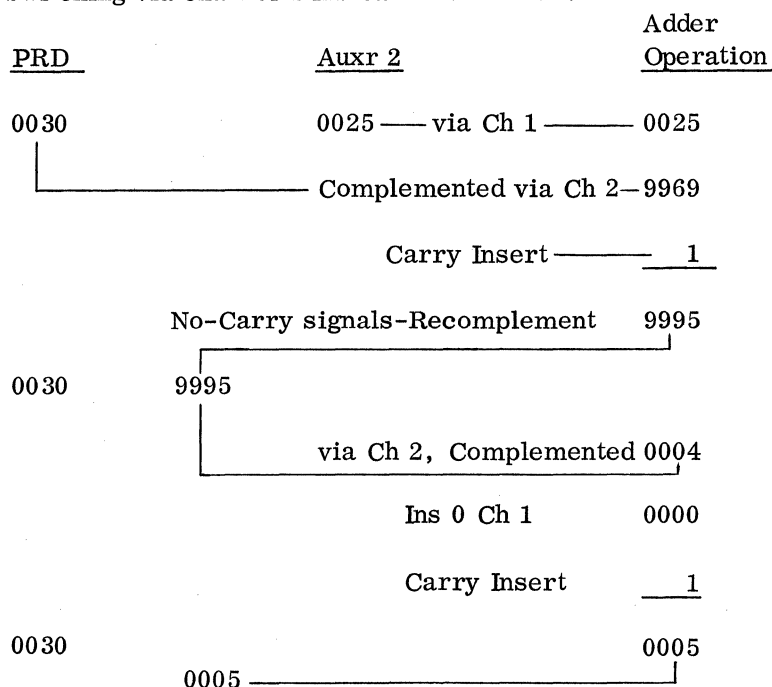


FIGURE 5.9-13. SEQUENCE FOR INDEX WORD ADD TO - SUBTRACT FROM INDEXING PORTION, OP $\pm$ 47

[illegible]

FIGURE 5.9-14. SEQUENCE FOR INDEX WORD ADD TO - SUBTRACT FROM INDEXING PORTION, COMPLEMENT ADD WITH RECOMPLEMENT

Complement Adjust if the Result Goes Through Zero. If a complement-add operation occurs on either Op ADI or SDI, the result may have to be recomplemented. A "No Carry 1st" output of the Adder at Prog Cycle Last + time signals the complement adjust operation. The following illustration shows the arithmetic principles involved in a complement-add operation followed by a recomplement or complement adjust cycle. Note that the negative result, requiring recomplementing, ends up in Auxiliary Register 2. To be recomplemented, the contents of Auxr 2 must be read out to the adder complement entry switching via channel 2 instead of channel 1.



As shown on Figure 5.9-14, the program ring is recycled and the following gates are turned on to complement the contents of Auxr 2.

1. Comp Add
2. Advance Auxr 2
3. RO Auxr 2, Channel 2
4. Auxr 2 RI Adder
5. Channel 1 Insert 0

Sign Change if the Result Goes Through Zero. Figure 5.9-12 shows the logic required to change the sign of the index word when the value of the word goes through zero and the recomplement cycle is taken. At Program Cycle Last + time of the recomplement operation, the auxiliary register sign is changed from + to -, or from - to +. If the sign was alpha, no change takes place.

Store the Index Word Back in Core Storage. At program cycle test time a conventional Store-in-Memory cycle is initiated. This occurs at the completion of the first adder cycle when no recomplementing is necessary, or, when the result of the 1st adder operation is negative, after the recomplement cycle. The modified index word is stored back in the address designated by the field register portion of the program register.

5.9.08 Index Word Set Non-Indexing Portion

Op + 48 Index Word Set Non-Indexing Portion (XSN)

The number in program register D replaces the non-indexing portion of the specified index word. The sign and the remaining positions of the index word are not changed.

Introduction and Circuits(Figures 5.9-15, 16)

The index word designated by the field address portion of the program register is read out to the auxiliary register by a normal data request operation. After the Op code signal, Op SDF, is developed, a store-in-memory operation is developed at C0. Auxiliary 1 and 2 are read out to the IB at Store Control 3, AP, time. The read out of Auxiliary 3, positions 6 - 9, is inhibited. Instead, program register D is read out to the IB. Thus, the revised index word is returned to the core storage location designated by the field address.

5.9.09 Index Word Load and Interchange

Op - 48 Index Word Load and Interchange (XLIN)

The contents of the core storage location or accumulator designated by the address in program register D replace the previous contents of the specified index word. Positions 2 - 5 and 6 - 9 are interchanged in the process as illustrated below.

Addressed core storage location	+ 00	0039	0348
Index Word designated by Fld Addr becomes	+ 00	0348	0039

Introduction and Circuits(Figures 5.9-17, 18)

Op - 48 is an initial access code in which the contents of the core storage location specified by the address in program register D are read out to the auxiliary register. The indexing and non-indexing portions of the index word in the auxiliary register are interchanged. The result is stored back in the index word location specified by the field address of the instruction.

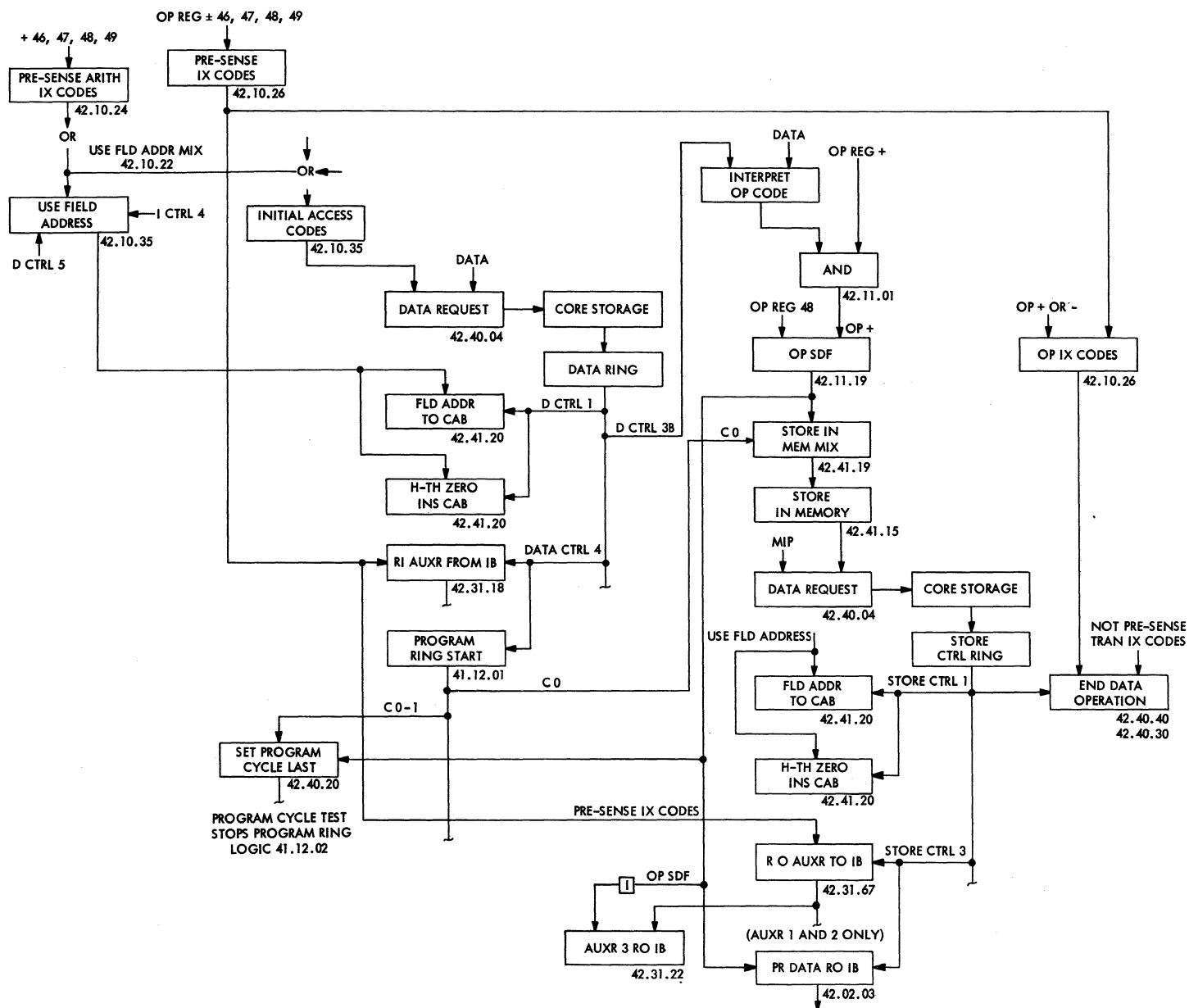
Figure 5.9-17 illustrates the manner in which the indexing portion is interchanged with the non-indexing portion. Auxiliary 2 is serially shifted to the right by the Advance Auxr 2 gate. The Auxr 2 digits are gated into Auxr 3 by the Auxr 3 RI Auxr 2. At the same time, Auxiliary 3 is shifted right by the Advance Auxr 3 signal and gated into the Auxr 2 high order end by Auxr 2 Regen RI 3. Next, a store in memory request is developed at C3 which stores the index word in the location specified by the PR field address.

5.9.10 Branch Incremented Index Word

Op + 49 Branch Incremented Index Word (BIX)

The absolute value of the indexing portion of the specified index word is increased by 1. This new indexing portion is then compared to the non-indexing portion of the same index word. If the absolute value of the new indexing portion is still not greater than the absolute value of the non-indexing portion, a branch to the data address of the PR occurs. When the indexing portion becomes greater, the next instruction is taken from the address indicated by the instruction counter.

FIGURE 5.9-15. INDEX WORD SET NON-INDEXING PORTION



MEMORY CYCLE MEMORY CLOCK		I	O	I	O	I	O	I	O	I	O	I	O	I	O
		UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ
PROGRAM CYCLE RING														Last +	Stop
DATA, STORE CONTROL RING					D1	D3A	D4				S1	S3	S4B	S6	
					D2	D3B	D5				S2	S4A	S5		
INSTRUCTION CONTROL RING		I3A	I4								I1	I3A	I4		
		I2	I3B	I5							I2	I3B	I5		
PROGRAM RING					CX-0						CX-0	C0-1	C1-2		
SIGNAL NAME		LOGIC	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD
PRE-SENSE ARITH I X CODES		42.10.24	.	.	.	.	.	.	.	.	.	.	.	.	.
INITIAL ACCESS CODES		42.10.19	.	.	.	.	.	.	.	.	.	.	.	.	.
		42.10.35	.	.	.	.	.	.	.	.	.	.	.	.	.
USE FIELD ADDRESS		42.10.22	.	.	.	.	.	.	.	.	.	.	.	.	.
		42.10.35	.	.	.	.	.	.	.	.	.	.	.	.	.
DATA		42.40.04	.	.	.	.	.	.	.	.	.	.	.	.	.
FIELD ADDRESS TO CAB		42.41.20	.	.	.	.	.	.	.	.	.	.	.	.	.
R I AUXR FROM I B		42.31.19	.	.	.	.	.	.	.	.	.	.	.	.	.
		42.31.18	.	.	.	.	.	.	.	.	.	.	.	.	.
OP SDF		42.11.19	.	.	.	.	.	.	.	.	.	.	.	.	.
STORE IN MEMORY MIX		42.41.19	.	.	.	.	.	.	.	.	.	.	.	.	.
STORE IN MEMORY		42.41.15	.	.	.	.	.	.	.	.	.	.	.	.	.
FIELD ADDRESS TO CAB		42.41.20	.	.	.	.	.	.	.	.	.	.	.	.	.
R O AUXR TO I B		42.31.67	.	.	.	.	.	.	.	.	.	.	.	.	.
PR DATA R O I B		42.02.03	.	.	.	.	.	.	.	.	.	.	.	.	.
		42.02.23	.	.	.	.	.	.	.	.	.	.	.	.	.
INH AUXR 3 R O I B		42.31.22	.	.	.	.	.	.	.	.	.	.	.	.	.
SET PROG CYCLE LAST		42.40.20	.	.	.	.	.	.	.	.	.	.	.	.	.
END DATA OPERATION		42.40.40	.	.	.	.	.	.	.	.	.	.	.	.	.
		42.40.30	.	.	.	.	.	.	.	.	.	.	.	.	.
INSTRUCTION		42.40.05	.	.	.	.	.	.	.	.	.	.	.	.	.
			.	.	.	.	.	.	.	.	.	.	.	.	.
			.	.	.	.	.	.	.	.	.	.	.	.	.
			.	.	.	.	.	.	.	.	.	.	.	.	.

FIGURE 5.9-16. SEQUENCE FOR INDEX WORD SET NON-INDEXING PORTION, OP + 48

The absolute value of the indexing portion of the specified index word is increased by 1. This new indexing portion is then compared to the non-indexing portion of the same index word. If the absolute value of the new indexing portion is still not greater than the absolute value of the non-indexing portion, a branch to the data address of the PR occurs. When the indexing portion becomes greater, the next instruction is taken from the address indicated by the instruction counter.

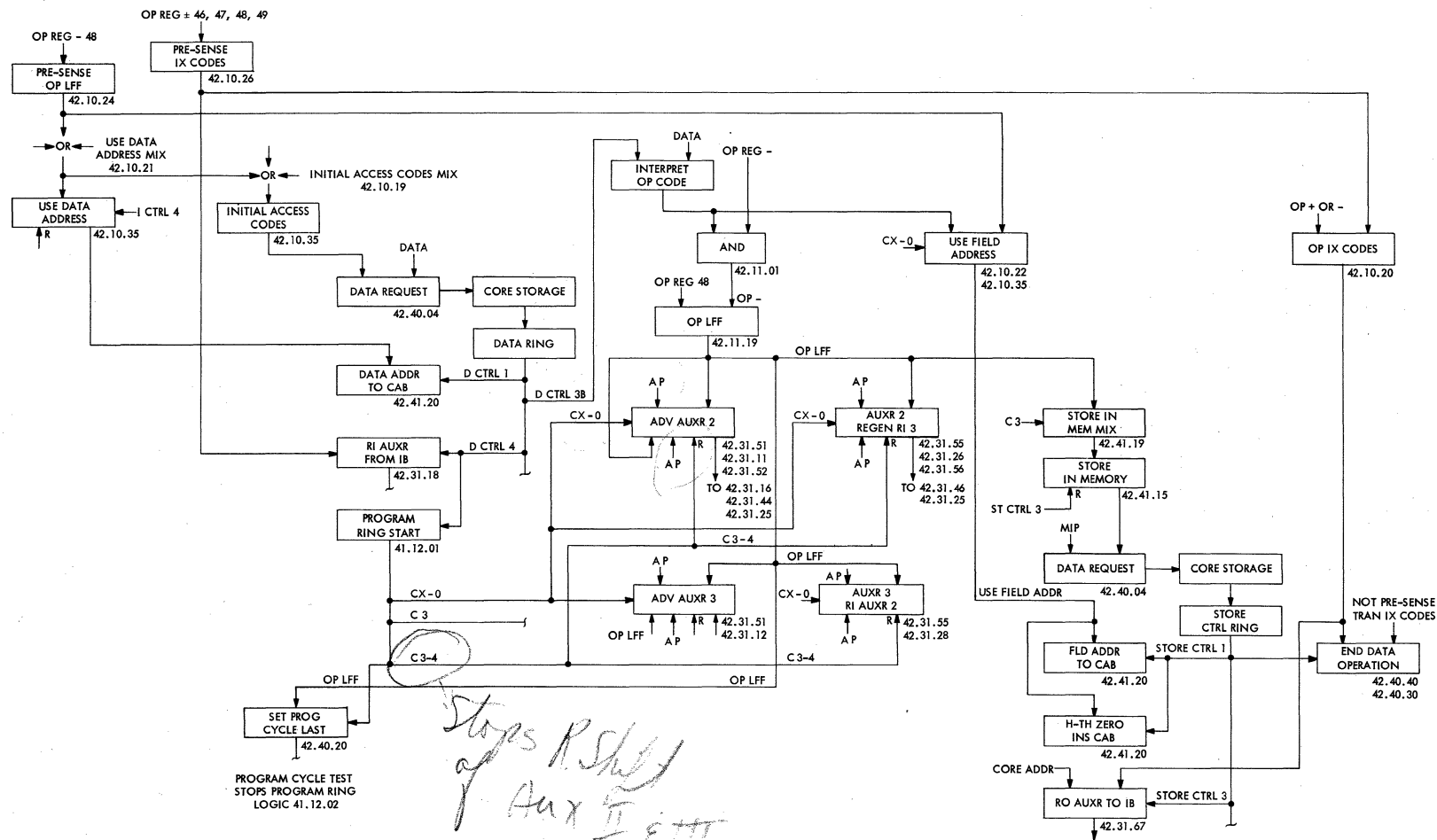
Circuits for Branch Incremented Index Word(Figures 5.9-19, 20)

The following objectives are accomplished to increase the value of the indexing portion by 1 and compare the indexing portion to the non-indexing portion:

Read out the Index Word from Core Storage. This is a standard memory access operation using the PR Field Address to address core storage. The designated index word is read into the auxiliary register as shown in Figure 5.9-19.

Add 1 to AUXR 2 (Indexing Portion). The operation control signals, Op Inc Tr and Op Inc Tr or Dec Tr (Figure 5.9-19) cause the following: Auxr 2 contents are read out to the adder on channel 2. At the same time, zeros are inserted on channel 1. The 1 is added by inserting a carry on the adder input. The adder output is then brought back to Auxr 2. The following signals are developed during C1 - C5 time to accomplish this objective (Figure 5.9-19, 20):

FIGURE 5.9-17. INDEX WORD LOAD AND INTERCHANGE, OP - 48



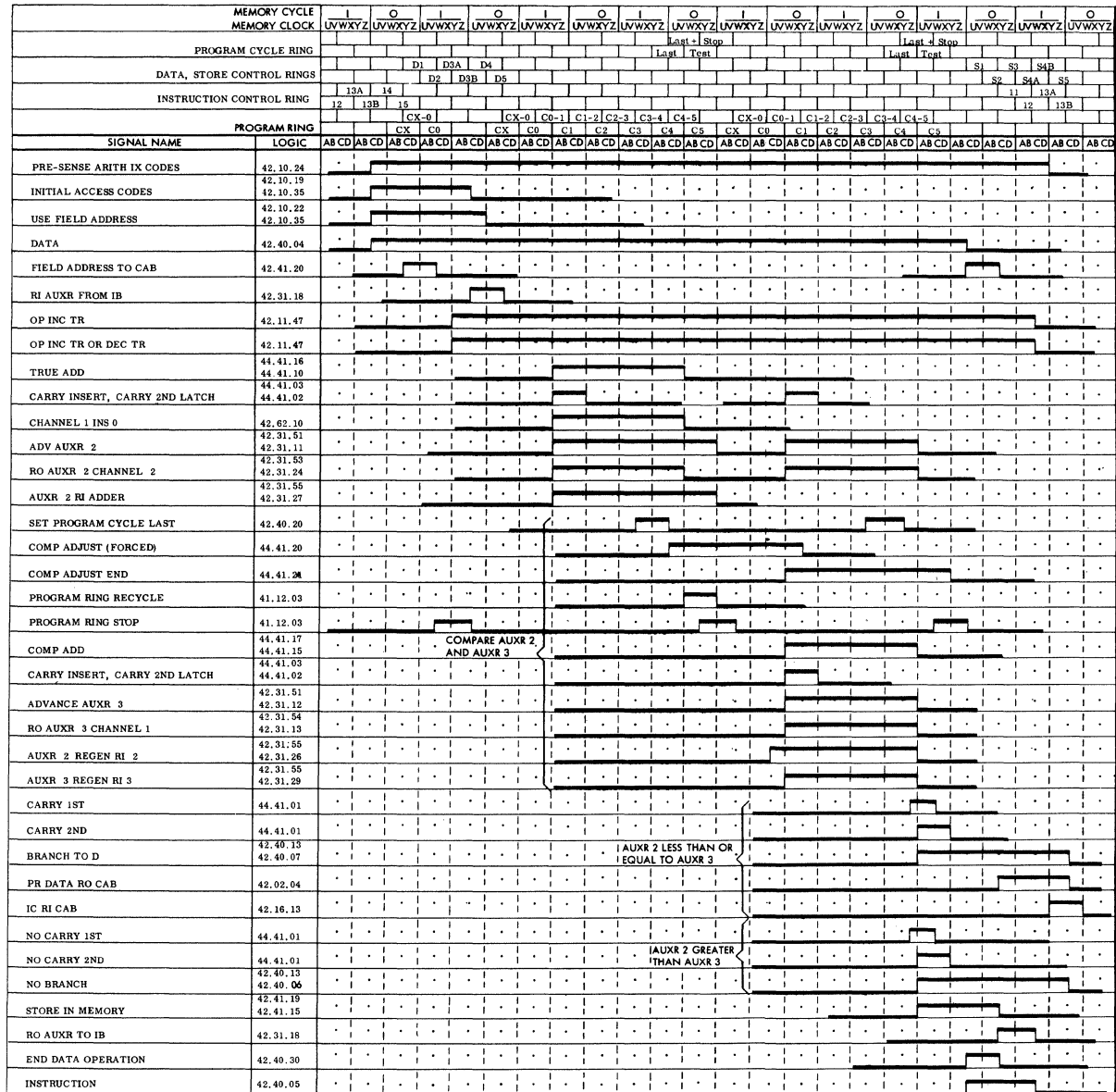


FIGURE 5.9-20. SEQUENCE FOR BRANCH INCREMENTED INDEX WORD, OP + 49

1. Comp Add
2. Advance Auxr 2
 - a. RO Auxr 2, Channel 2
 - b. Auxr 2 Regen RI 2
3. Advance Auxr 3
 - a. RO Auxr 3, Channel 1
 - b. Auxr 3 Regen RI 3
4. Carry Insert

Note that the contents of both Auxr 2 and 3 enter the adder and also are regenerated. Thus, the auxiliary register contents at the end of the complement operation are the same as before.

Test for Branch or No Branch. As long as the non-indexing portion of the index word is greater than the indexing portion, a Branch-to-D results. The following arithmetic operation occurs:

Non-Indexing	0625	to Adder A Entry	Machine Complementing 0625
Indexing	0623	to Adder B Entry	<u>9377</u>
			←C 0002

A carry from the high-order end indicates that the Non-indexing portion is larger and a Branch-to-D signal results at the end of the Comp Adjust cycle (Figure 5.9-19).

Non-Indexing	0625	to Adder A Entry	Machine Complementing 0625
Indexing	0626	to Adder B Entry	<u>9374</u>
			←NC 9999

A no carry from the high-order end indicates that the indexing portion has become greater than the non-indexing portion and a No-Branch results (Figure 5.9-19). The IC address is used for the next instruction.

Store the Revised Index Word in Core Storage. This is a conventional store-in-memory operation using the field address of the instruction to address core storage. It is signalled at the end of the complement-adjust operation by a program cycle test gate.

5.9.11 Branch Decrement Index Word

Op - 49 Branch Decrement Index Word (BDX)

The non-indexing portion of the specified index word is subtracted from the indexing portion. The result replaces the previous indexing portion and the index word is stored back in core storage. The sign of the result becomes the sign of the index word.

If the subtraction causes a zero result or a sign change, a No-Branch occurs and the next instruction is taken from the address specified by the instruction counter. If no sign change or zero occurs, a Branch-to-D is signalled and the next instruction is taken from the location specified by the address in program register D.

Introduction

Op Code - 49 is an initial access code in which the index word to be decremented is read out from core storage to the auxiliary register. The contents of Auxr 3 (non-indexing portion) is complement added to the contents of Auxr 2 (indexing portion). The three possible results of this operation are shown in the following summary:

Index Word	Before	+ 00	0002	0001
	After	+ 00	0001	0001

Arithmetic Operation: 0002

9999

← C | 0001

Results:	Carry	}	Branch to D
	Adder Non-Zero		

Index Word	Before	+ 00	0001	0001
	After	+ 00	0000	0001

Arithmetic Operation: 0001

9999

← C | 0000

Results:	Carry	}	No Branch
	Adder Zero		

Index Word	Before	+ 00	0000	0001
	After	- 00	0001	0001

Arithmetic Operation: 0000

9999

← NC | 9999

Results:	<u>No Carry</u>	}	No Branch
	Adder Non-Zero		

Sign Change-Comp. Adj.

Figure 5.9-21 shows a simplified block diagram of the operation, indicating how the adder output is tested to result in either Branch-to-D or No-Branch with a complement adjust taking place whenever a No-Carry indicates that the sign has changed from + to -. If a complement adjust operation is necessary, the program ring is recycled and the usual complement adjust takes place.

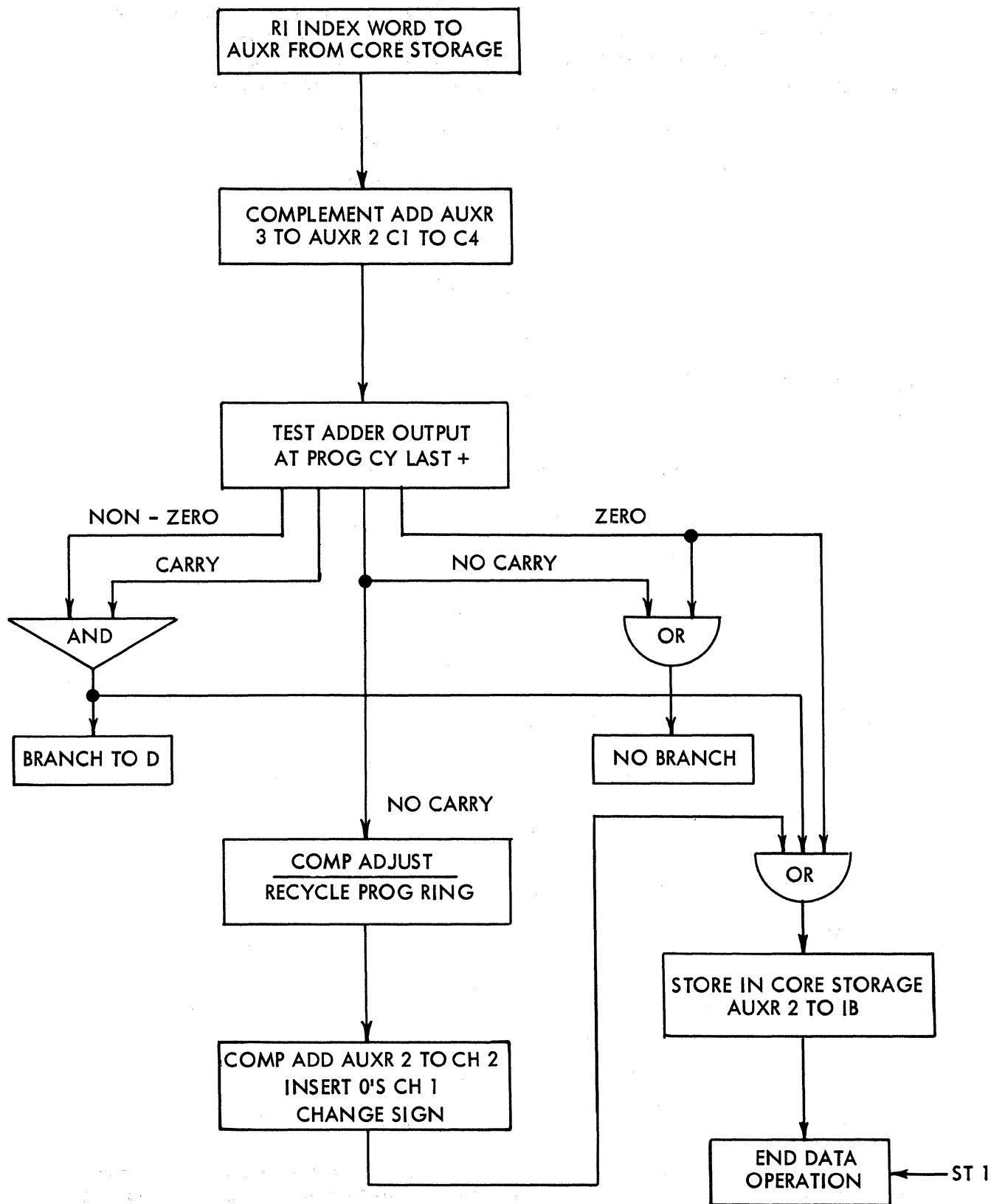


FIGURE 5.9-21. BRANCH DECREMENTED INDEX WORD BLOCK DIAGRAM

Circuits for Branch Decrement Index Word, Dec Tr

Figure 5.9-22 outlines the circuits required. The sequence of operation is shown in Figure 5.9-23. The following major objectives must be accomplished:

Read Out Index Word from Core Storage. This is a standard memory access operation using the PR field address. The designated index word is read into the auxiliary register as shown in Figure 5.9-22.

Subtract AUXR 3 from Auxr 2. The operation control signals, Op Dec Tr and Op Inc Tr or Dec Tr (Figure 5.9-22) cause the following gates which control the data flow from Auxr 2 to the Adder and from Auxr 3 to the adder:

1. Comp Add
2. Advance AUXR 2
 - a. Auxr 2 RO to Ch 1
 - b. Auxr 2 RI Adder
3. Advance AUXR 3
 - a. Auxr 3 RO Ch 2
 - b. Auxr 3 Regen RI 3
4. Carry Insert

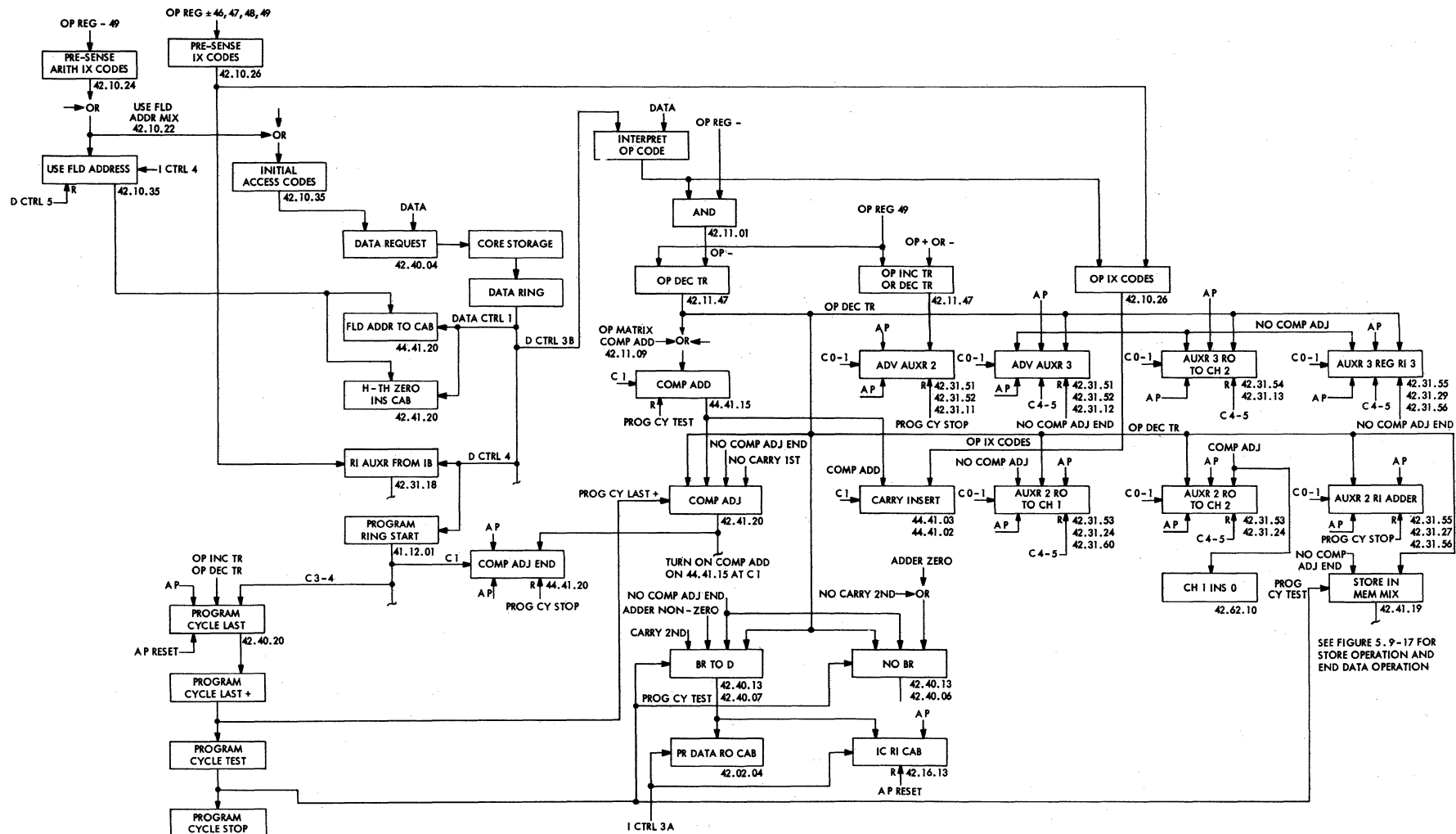
Test for Branch or No Branch. The general requirements for the Branch and No-Branch Test are discussed in the introduction. Circuits are shown in Figure 5.9-22.

Complement Adjust Auxr 2 in Case of Sign Change. When a no-carry from the adder indicates that the sign has changed from + to -, a complement-adjust cycle is started. In addition to the gates shown at the bottom of the sequence chart (Figure 5.9-23), the following are required to control the data flow to the adder (See Figure 5.9-22 for locations and timing):

1. Advance Auxr 2
2. Auxr 2 RO Ch 2
3. Auxr 2 Regen RI Adder
4. Insert Zeros Ch 1
5. Carry Insert

Store the Revised Index Word in Core Storage. This is a conventional store-in-memory operation, using the field address of the instruction to address core storage. It is signalled at the end of the decrementing operation by a program cycle test gate. In the case of a complement-adjust operation, the store operation is delayed by switching No Comp Adj with Op Dec Tr (Figure 5.9-22) so that the store operation does not occur until the Prog Cycle Test gate at the end of Comp Adjust.

FIGURE 5.9-22. BRANCH DECREMENTED INDEX WORD, OP - 49



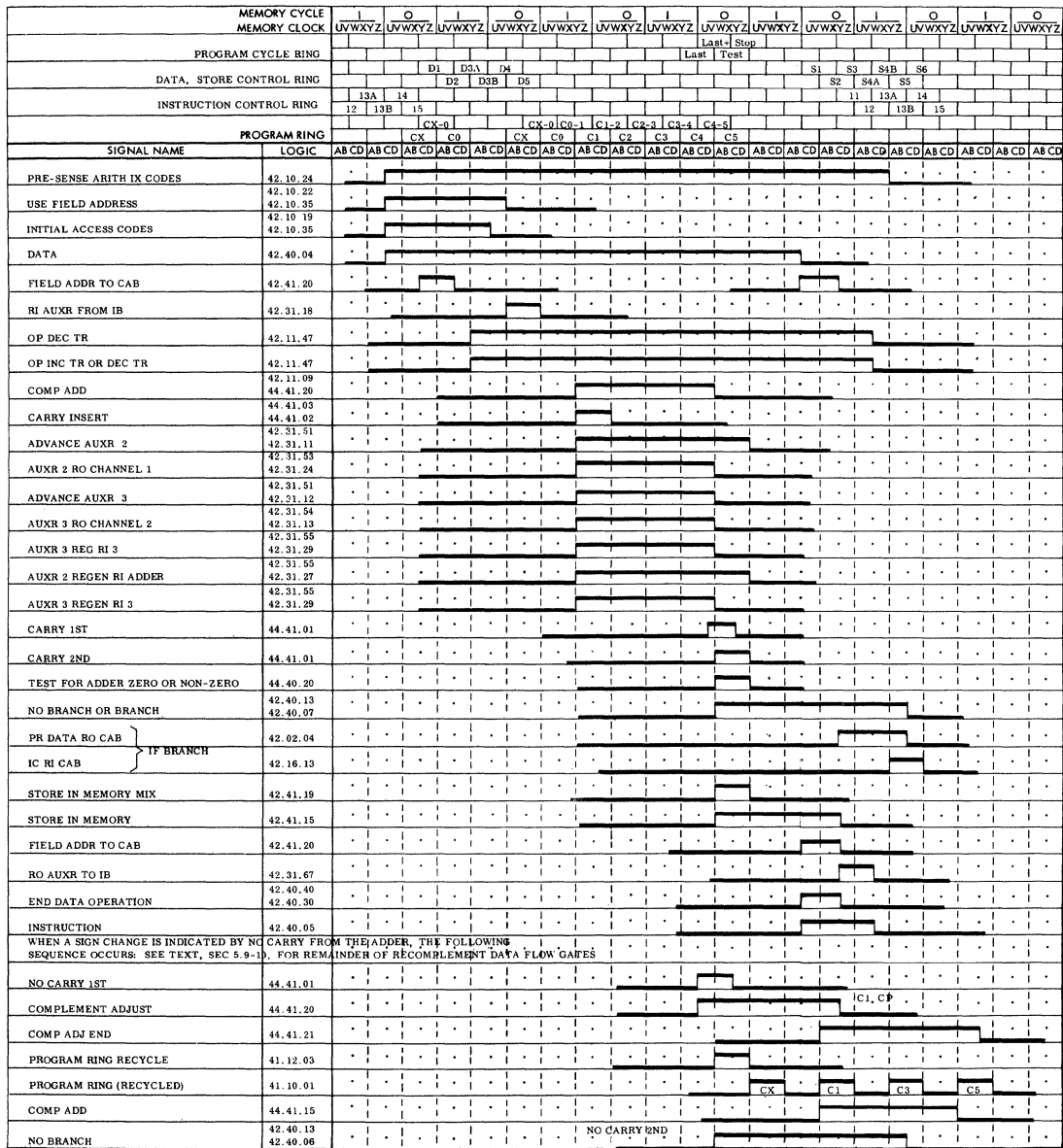


FIGURE 5.9-23. SEQUENCE FOR BRANCH DECREMENTED INDEX WORD, COMP ADD, NO SIGN CHANGE, NO COMP ADJUST

5.10.00 TABLE LOOKUP

Three different types of table searches can be performed as follows:

1. TLU Equal. A search for an entry in the stored table equal to the value specified (Argument).
2. TLU Equal or High. A search for an entry equal to, or greater than, the argument.
3. TLU Lowest. A search to determine the lowest entry in the complete table.

The argument for the table lookup must be placed in accumulator 3 before the look-up instruction is performed. This number may be of any length up to 10 digits. Positions 4 and 5 of the instruction (Field definition) specify the field within the table words that is to be compared to the argument.

The location of the table in core storage is specified by record definition words (also called record control words). The location of the first record definition word is specified by positions 6 - 9 of the TLU instruction. The search continues until the required conditions are met or until the last record definition word (identified by a minus sign) has been used. On TLU lowest, the search always continues until the last record definition word is used.

The working address for the table lookup may be incremented by one (as in tape operations) or by any other fixed amount. The non-indexing portion of word 0098 is used to store the increment. The increment is considered +, regardless of the sign of the indexing word.

For example, assume the following TLU instruction and record definition words:

	S	0	1	2	3	4	5	6	7	8	9
TLU Inst	+	6	7	0	0	0	9	0	1	0	5
Location											
0 1 0 5	+00		0	5	4	1					0
0 1 0 6	+00		0	4	2	0					0
0 1 0 7	-00		1	9	9	0					1
0 0 9 8	+00		X	X	X	X					0

Record Definition Words

With the preceding series of words stored in core storage, a TLU Equal operation is performed. The full 10-digit field of every second word in the table is compared to the 10 digit number (and sign) previously placed in Accumulator 3. When an equal value is found, the address of the equal value is placed in positions 2 - 5 (XXXX) of index word 98. The record control words located in 0105, 6, 7 define the table from 0541 to 0560, from 0420 to 0463, and from 1990 to 1999 (inclusive in each case) which is to be searched. Only the values in 0541, 0543, 0545 ; 0420, 0422etc. are compared.

5.10.01 TLU Equal, and Equal or High

Op + 67 Lookup Equal Only (LE)

If an entry in the table to be searched is found to be equal to the argument in accumulator 3, its address replaces the indexing portion of index word 98. The lookup ends, and the next instruction in order is skipped. If an entry equal to the argument is not found, index word 98 is not changed and the next instruction in order is executed.

Op + 68 Lookup Equal or High (LEH)

If an entry in the table is found to be equal or higher than the argument, its address replaces the indexing portion of index word 98. The lookup ends and the next instruction in order is skipped. If an entry satisfying the equal or higher condition is not found, index word 98 is not changed, and the next instruction in sequence is executed.

Introduction

Table Lookup Equal, and Equal or High are very similar in function, the major difference being the switching required to indicate the "find" condition when the argument in Accumulator 3 is compared to the word from core storage in the Athr. The two codes are discussed as one subject and the slight differences are pointed out. Figures 5.10-1 and 5.10-2 show flow charts of both operations. Flow charts are keyed to both circuit logic and sequence charts by use of the symbols  and . For example, to find both circuits and timing for Comp Add Auxr 2 to Auxr 3  on Figure 5.10-1 see circuits labelled  on 5.10-4A and the sequence of operation  on Figure 5.10-6A.

The Table Lookup Operation Flow charts, Figures 5.10-1 and 2, outline the major steps involved. The following comments apply to the numbered blocks.

-  After the TLU instruction has been transferred to the PR and the IC is One-upped by the normal instruction cycle routine, the address of the first record control word (RCW) is transferred from program register D to the address control A register. This becomes the storage register location for the next RCW for the TLU operation.
-  A normal memory access operation brings positions 6 - 9 of word 0098 to PR D to replace the address of the first RCW. The address 0098 is generated. This operation places the increment, to be used in increasing the start address of the RCW, in program register D.
-  A normal memory access operation brings the first RCW from the address specified by the address control A register to the auxiliary register.
-  The One-up circuits used to increment the IC are used to add one to the contents of the Address Control A Register. Thus, the address of the next RCW is placed in the Addr Ctrl A Reg for later use.
-  Before proceeding with the actual TLU operation, the following conditions are checked:
 - a. Is the start address equal to or less than the stop address? If it is, at least one TLU operation is performed.

* O

FIGURE 5.10-1. TLU EQUAL OPERATION FLOW CHART

- b.

Add operation on Step (12).

* ○ REFER TO FIGURE 5.10-4

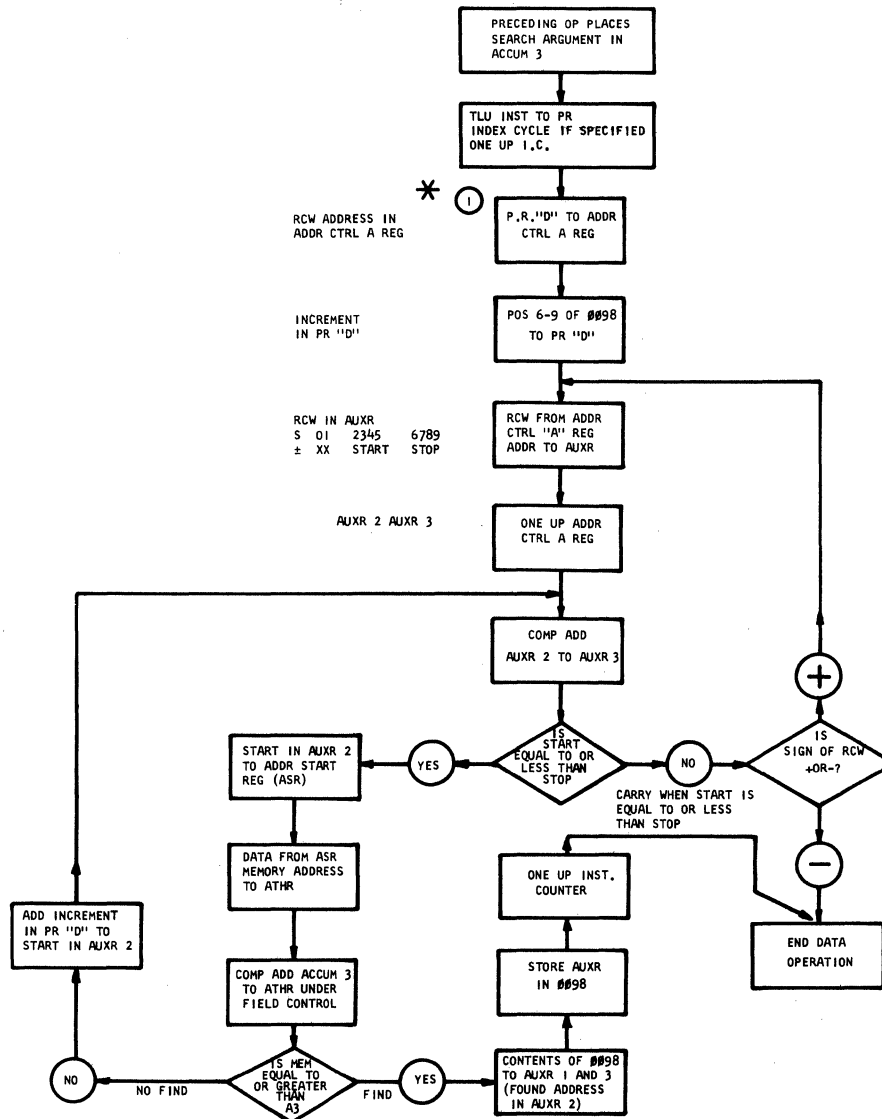


FIGURE 5.10-2. TLU EQUAL OR HIGH OPERATION FLOW CHART

- REFER TO FIGURE 5.10-4
 □ REFER TO FIGURE 5.10-5

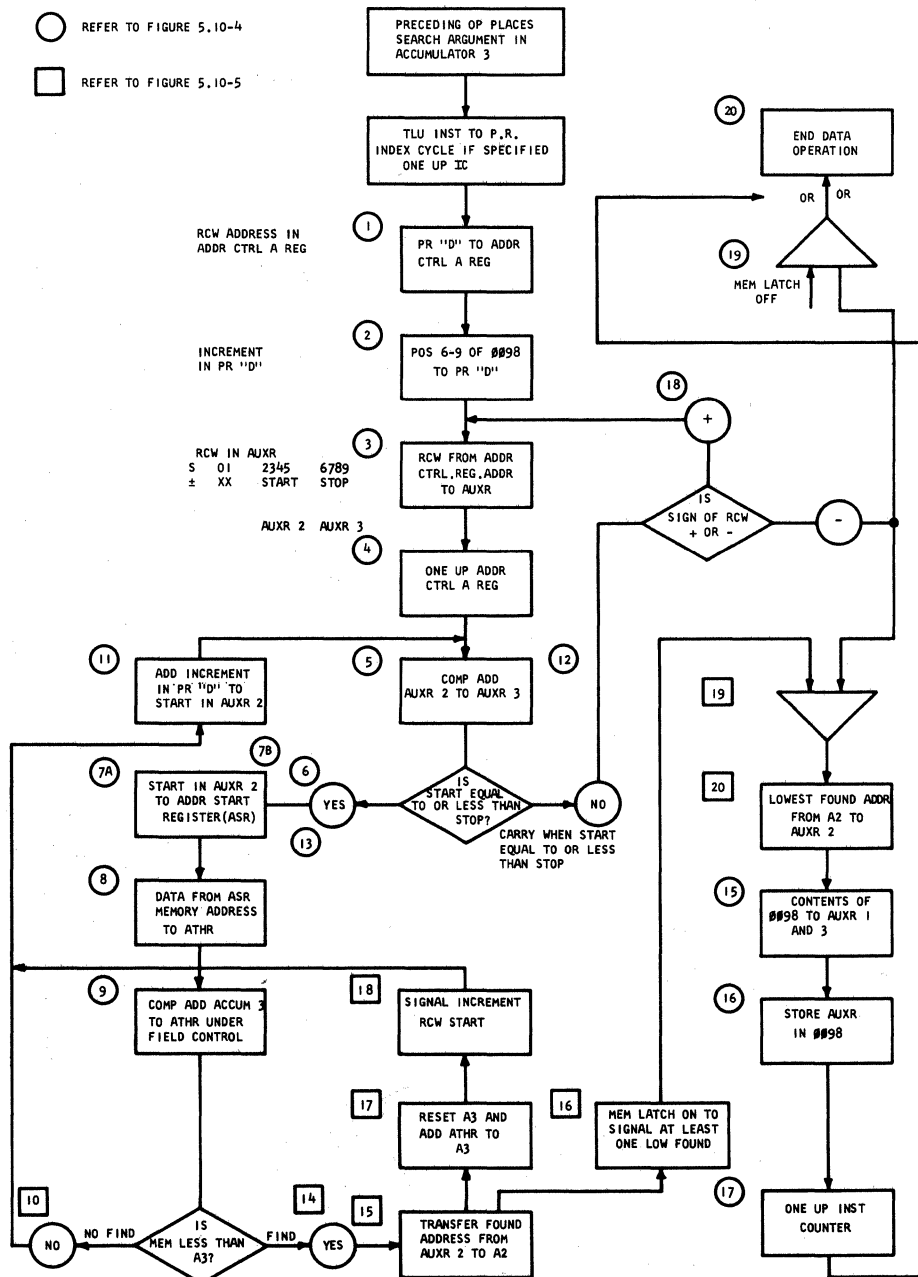


FIGURE 5.10-3. TLU LOWEST OPERATION FLOW CHART

		Address	Arith Op
Auxr 3	STOP	0 0 0 8	0 0 0 8
Auxr 2	START	0 0 0 6	9 9 9 3
			_____ 1 (Carry Insert)
Carry indicates START equal	← C		0 0 0 2
to or less than STOP			

		Address	Arith Op
Auxr 3	STOP	0 0 0 8	0 0 0 8
Auxr 2	START	0 0 0 9	9 9 9 0
			_____ 1 (Carry Insert)
No-Carry indicates START	← NC		9 9 9 9
greater than STOP			

On the first test, Tran Ctrl 2 switches with No-Carry to cause a match error. On the second and additional tests, No-Carry switches with the RCW sign to cause either the read in of a new RCW (+ sign) or an end data operation (- sign).

- ⑥ Assuming that an adder carry results from the comparison of ⑤, a continue-lookup signal is generated (Figure 5.10-4A) which signals step ⑦A.
- ⑦A The next Start address is transferred from Auxr 2 to the Address Start Register (ASR).
- ⑧ The core storage data to be compared to the argument in Accumulator 3 is transferred to the Athr by a memory access operation which uses the Start address in the ASR to address core storage.
- ⑨ This is the actual comparison operation which complement adds the argument in A3 to the factor from core storage in the Athr. The adder output signals are tested at the end of the operation for carry/no-carry and zero/non-zero to determine whether a Find for the particular operation code has been attained. The following arithmetic operations may occur (Note that Find conditions are keyed to Figure 5.10-1, 2, 3, by ⑩ and No-Find by ⑭ and ⑭):

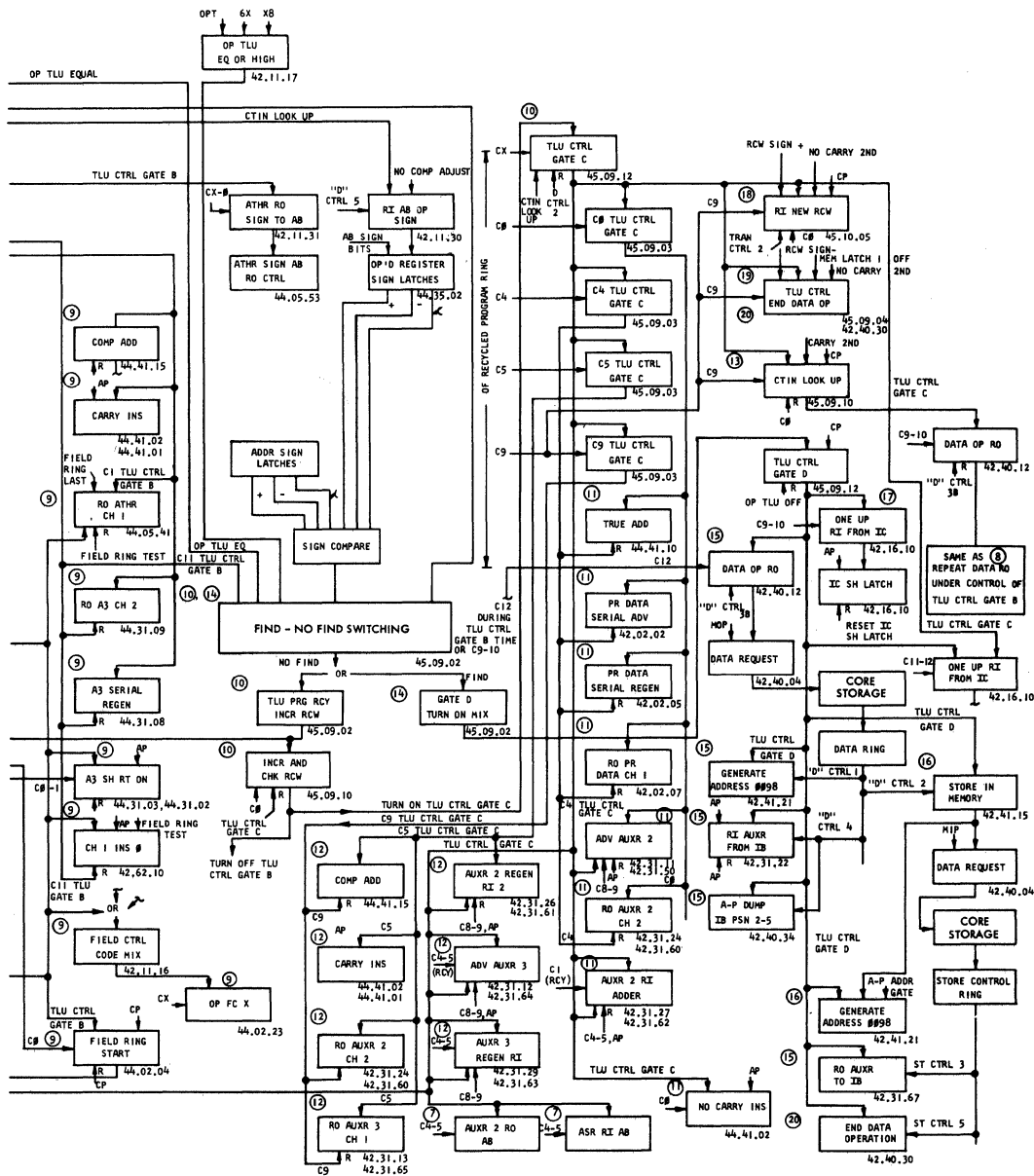
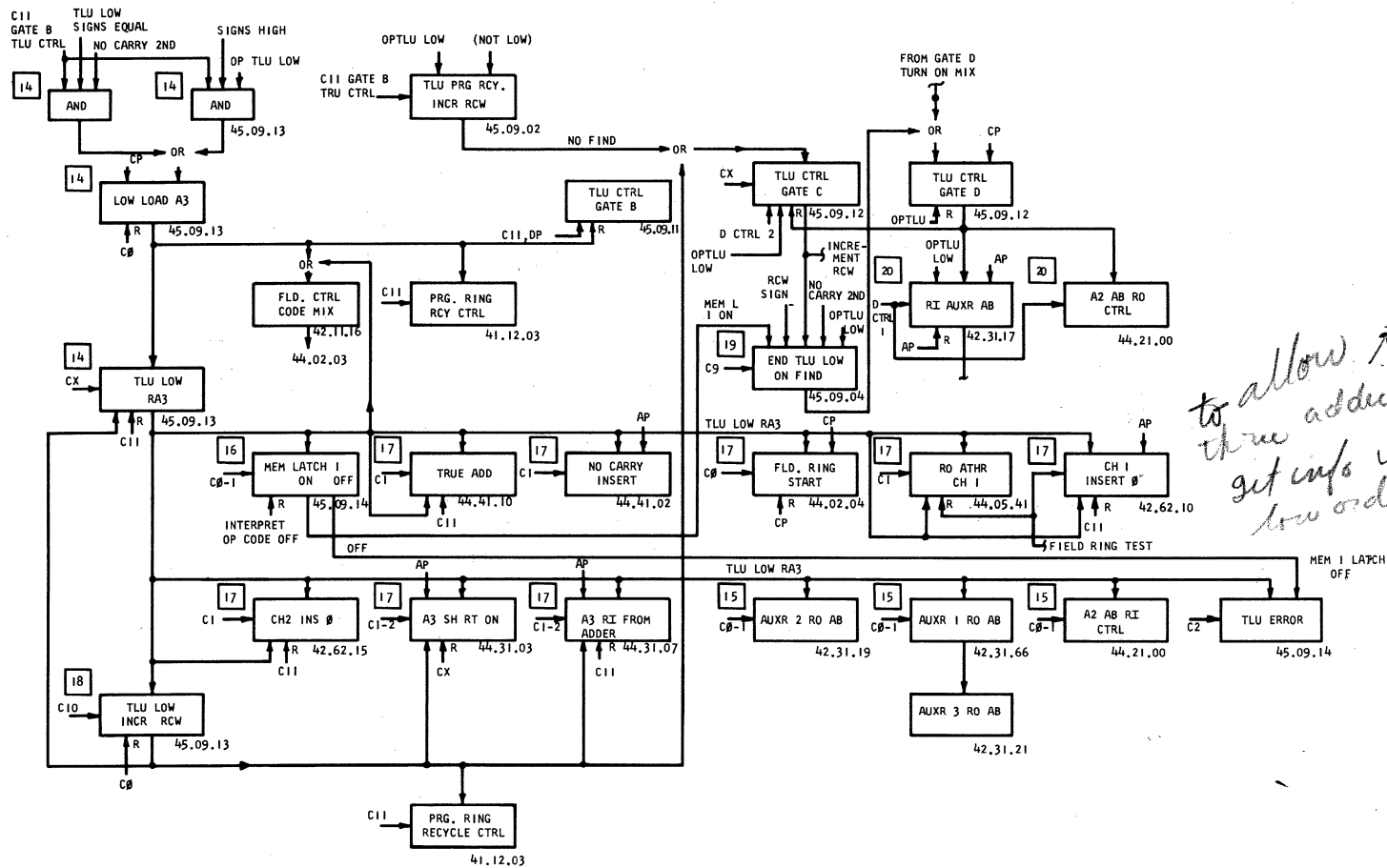


FIGURE 5.10-4B. TLU EQUAL LOGIC

FIGURE 5.10-5. ADDITIONAL LOGIC FOR OP TLU LOWEST



*to allow running with
three addres to be able to
get info into acc III
low order positions.*

Signs Equal + or alpha

Register
Contents

Arithmetic
Operation

Athr

0 0 3

0 0 3

A3

0 0 3

9 9 6

1 Carry Insert

C

0 0 0

- ⑭ On TLU Equal, Adder 0 indicates find.
- ⑭ On TLU Equal or High, Carry indicates find.
- ⑩ On TLU Lowest, Carry indicates no find.

Register
Contents

Arithmetic
Operation

Athr

0 0 2

0 0 2

A3

0 0 3

9 9 6

1 Carry Insert

NC

9 9 9

- ⑩ On TLU Equal, Adder non-zero indicates no find.
- ⑩ On TLU Equal or High, No-Carry indicates no find.
- ⑭ On TLU Lowest, No-Carry indicates find.

Signs Equal and Minus.

Register
Contents

Arithmetic
Operation

Athr

(-) 0 0 4

0 0 4

A3

(-) 0 0 3

9 9 6

1 Carry Insert

C

0 0 1

- ⑩ On TLU Equal, Adder non-zero indicates no find.
- ⑩ On TLU Equal or High, Adder non-zero and carry indicates no find.
- ⑭ On TLU Lowest, Carry and Adder non-zero indicate find.

Register
Contents

Arithmetic
Operation

Athr

(-) 0 0 2

0 0 2

A3

(-) 0 0 3

9 9 6

1 Carry Insert

NC

9 9 9

- ⑩ On TLU Equal, Adder non-zero indicates no find.
- ⑭ On TLU Equal or High, No-Carry indicates find.
- ⑩ On TLU Lowest, No-Carry indicates no find.

	Register Contents	Arithmetic Operation
Athr	(-) 0 0 3	0 0 3
A3	(-) 0 0 3	9 9 6
		<u>1</u> Carry Insert
		0 0 0

← C

- ⑭ On TLU Equal, Adder zero indicates find.
- ⑭ On TLU Equal or High, Adder zero indicates find.
- ⑩ On TLU Lowest, Adder zero indicates no find.

When signs are not equal, the find or no find condition is immediately determined without considering the adder output conditions. For example, a signs-high signal indicates that the sign of the argument (in A3) is higher than the sign of the factor from core storage. All of the following combination result in a signs-high signal.

A3 + - +

or or

Athr α α -

When TLU Equal or High is switched with the signs-high signal, a No-Find is indicated. Similarly, when TLU Equal or High is switched with the signs-low signal, a Find condition is indicated. In the same manner, Op TLU Lowest switched with signs-low results in a No-Find signal.

- ⑪ The increment, previously placed in PR D, is added to the last Start address used in Auxr 2. The result is returned to Auxr 2 to serve as the next TLU core storage address.
- ⑫ Immediately following step ⑪, a comparison of the start versus stop address is made to determine whether the start is equal to or less than stop. Refer to ⑤ for additional discussion of this step.
- ⑬ Carry, as a result of the Comp Add Operation in ⑫, signals a repetition of the TLU operation (Steps 7b, ⑧, ⑨, ⑩, ⑪, ⑫, ⑬).
- ⑭⑮ Whenever a Find is signalled by any of the conditions developed under ⑨ the address of the find must be transferred from Auxr 2 to the indexing portion of word 0098. To accomplish this, the contents of 0098 (except indexing portion which is dumped) are brought out to the auxiliary register by a memory access operation. The address 0098 is generated to address core storage.

- ①⑥ A store-in-memory operation transfers the revised word 0098 from the auxiliary register to the generated address 0098 in core storage.
- ①⑦ Because a find condition has been attained, the next instruction in order must be skipped. An additional One-up Instruction Counter operation is signalled to accomplish this.
- ②⑩ End Data Operation is signalled and a new instruction is read in to the PR.
- ①⑧ If on test ⑫, a no-carry occurs signifying that the RCW Start address is greater than the Stop address, a new RCW must be read in if the sign of the RCW is plus. Steps ③, ④, and ⑤ are repeated to bring a new RCW to the auxiliary register, one-up the Addr Ctrl A Reg to the new RCW address, and Test the RCW for Start equal to or less than Stop.
- ①⑨, ②① When No-Find has been attained and the sign of the RCW just used is minus, an end-data operation is signalled.

Circuits for Op TLU Equal and Op TLU Equal or High

The circuit logic for these Op codes is shown in Figures 5.10-4A and B. The circuits follow directly from the objectives outlined in the discussion above. As an additional guide, the TLU control gates are listed below along with the functions which they control. See Figures 5.10-6A, B and C for the sequence of operation.

Initial Access Codes. Signal memory access to transfer word 0098, pos. 6 - 9, to program register D.

TLU Control Gate A. Set up TLU Operation.

- A. ① Transfer of first RCW from PR D to Addr Ctrl Reg.
- B. ② Read in the increment from pos 6 - 9 of 0098 to program register D.
- C. Start memory access to get first RCW to Auxr.
- D. ③ Set RI New RCW latch to control core storage addressing.

Transfer Control 2. Control RI and initial Test of each RCW

- A. ③ Control RI of first RCW to the auxiliary register.
- B. ④ Signal One-up of Addr Ctrl Reg to next RCW Addr.
- C. ⑤ Complement add Auxr 2 to Auxr 2, Regen Auxr 2 and 3.
- D. ⑥ 1) Signal Continue Lookup if Carry indicates that Start is equal to or less than Stop.
2) Signal Match Error if No-Carry indicates that Start is greater than Stop.
- E. ⑦ Transfer first "Start" address from Auxr 2 to ASR.
- F. Start the memory access operation to read out the first table entry to the Athr.

TLU Control Gate B. Compare A3 to table value

- A. ⑧ Control RI of each table entry to the arithmetic register.
- B. ⑨ Set up gates required to Comp Add A3 to Athr under Field Control. A3 is regenerated.

FIGURE 5.10-6A. TLU SEQUENCE OF OPERATION

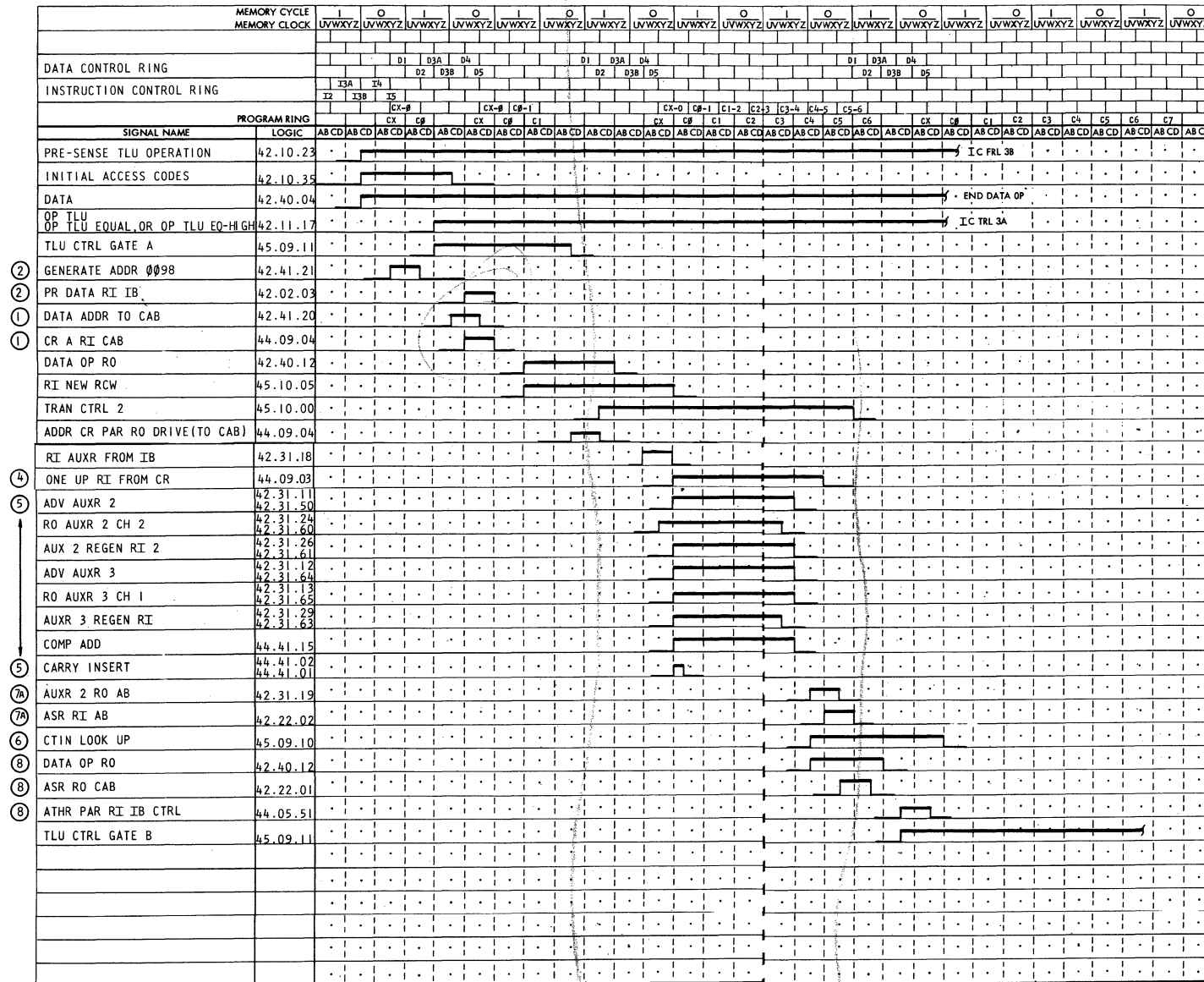
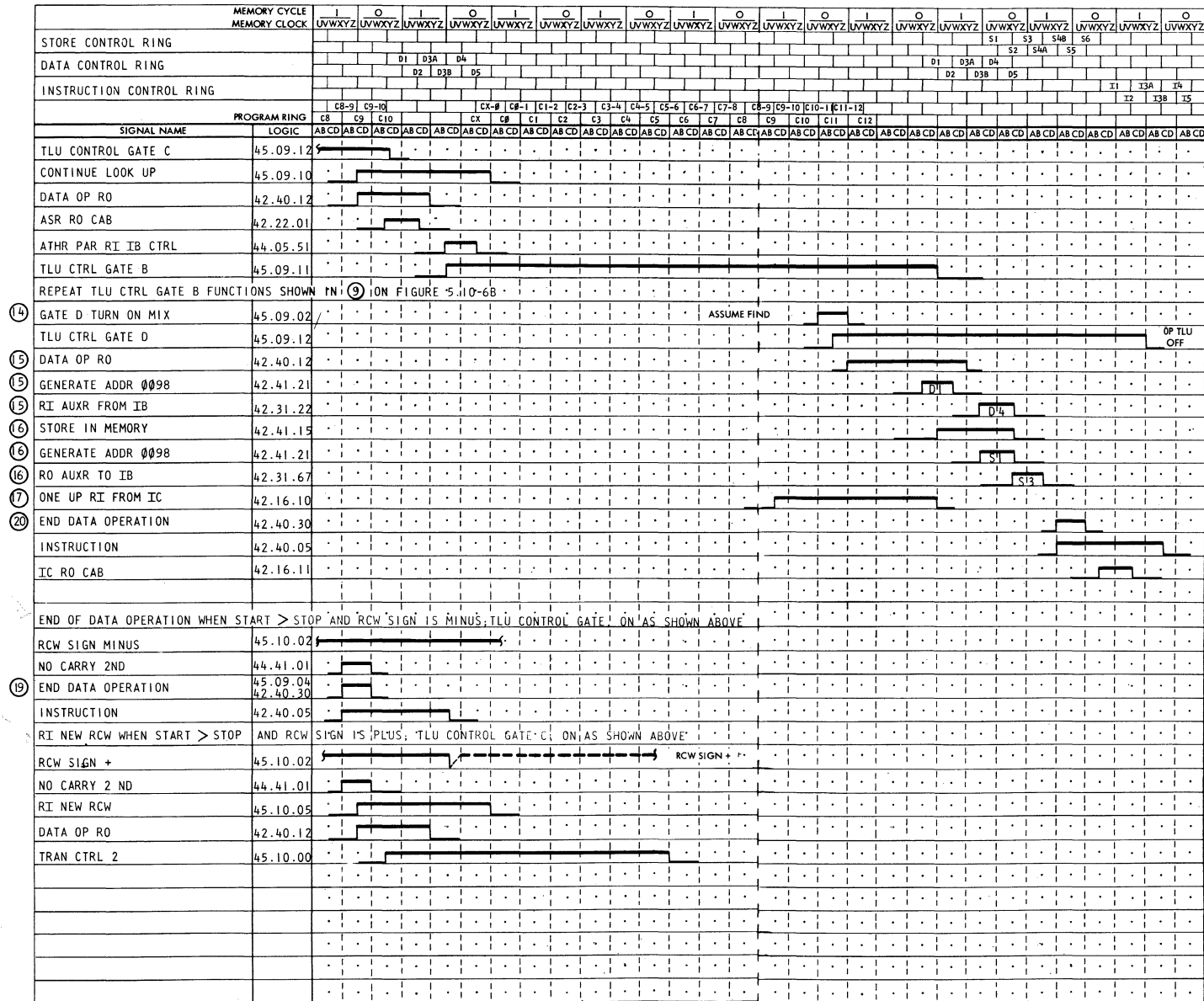


FIGURE 5.10-6C. TLU SEQUENCE OF OPERATION



- C. Transfer the Athr sign to the address sign latches in preparation for the C11 time find and no-find test.
- D. ⑩⑭ A C11 TLU Control Gate B, at the end of the Comp Add Operation of ⑨, tests for either a find or no-find condition. If a No-Find condition exists, the program ring is recycled and TLU Control Gate C is turned on to cause the incrementing of the Start address in Auxr 2. If a Find condition exists, TLU Control Gate D is turned on to cause the storage of the found address in 0098 and an end-data operation.

TLU Control Gate C. On with No Find-Increment Start and Test Start versus Stop.

- A. ⑪ Add the increment in PR D to the last used Start Address in Auxr 2.
- B. ⑫ Comp Add Auxr 2 to Auxr 3 to test for Start equal to or less than Stop.
- C. Test for the following as a result of the Comp Add of ⑫:
 - ⑬ 1) Continue Lookup if Carry indicates that Start is equal to or less than Stop.
 - ⑬ 2) RI New RCW if No-Carry indicates that Start is greater than Stop and RCW sign is plus.
 - ⑬ 3) End Data Operation if No-Carry indicates that Start is greater than Stop and RCW sign is minus.

TLU Control Word D. On with Find-Store "found" address.

- A. ⑮ Control the memory access operation by which the contents of 0098 are brought to Auxr 1 and 3. Positions 2 - 5 of 0098 are dumped since the found address occupies these positions in Auxr 2.
- B. ⑯ Signal the store-in-memory operation by which the auxiliary register contents are returned to core storage location 0098.
- C. ⑰ End-data operation as a result of the Find at Store Control 5 time.

5.10.02 TLU Lowest

Op + 66 Lookup Lowest (LL)

The search begins at the starting address of the first record-definition word and continues until the end of the table is reached. When any value lower than the argument in Accumulator 3 is found, the low value is placed in Accumulator 3 and the core storage address at which it was found is placed in Accumulator 2. This process continues until the end of the table is reached at which time, the address of the lowest factor in the table is stored in the indexing portion of word 0098.

The instruction next in sequence is skipped if at least one value lower than the original argument is found. If no value lower than the original argument is found, the next instruction in sequence is executed and neither accumulator 1, 2, 3, nor word 0098 are changed.

Introduction

The TLU Lowest Flow Chart (Figure 5.10-3) shows the steps of the TLU Lowest operation. The procedure is the same as other TLU Op codes until a find is signalled. At this point the following sequence takes place:

- [14] Arithmetic conditions which result in Find conditions on TLU Lowest are discussed in the introduction section of 5.10.01.
- [15] The found address is transferred from Auxr 2 to A2.
- [16] Mem latch 1 is turned on so that when the last RCW is used (RCW sign -), the address of the lowest found word can be transferred from A2 to Auxr 2 and then to the indexing portion of word 0098.
- [17] Reset the old lowest argument in A3 and transfer the last "found" lowest table word from the Athr to A3.
- [18] Signal return to the usual TLU routine at the point of incrementing the Start address in Auxr 2 by the contents of PR D procedure.

The procedure described above continues until step ⑫ determines that the Start address is greater than the Stop address. If the RCW sign is +, a new RCW is brought to the auxiliary register. If the RCW is minus, a test is made as follows:

- [19] If no table entry lower than the argument has been found, the Mem latch 1 is off and an immediate end-data operation is signalled and the next instruction in sequence is executed.
- [19] If at least one entry lower than the argument has been found, Mem latch 1 is On and the following operations occur:
- [20] The address of the lowest table entry is parallel transferred from Accumulator 2 to Auxiliary 2.
- ⑮,⑯ The normal store-in-memory operations are performed to place the lowest "found" address in the indexing portion of word 0098.
- ⑰ The IC is One-upped so that the next instruction in sequence is skipped.
- ⑳ End-data operation causes read-in of next instruction.

Circuits for Op TLU Lowest

Figure 5.10-5 shows the additional logic circuits required to accomplish steps [14], [15], [16], and [17] during a TLU Lowest operation. See Figures 5.10-7A and B for the sequence of operation. Whenever a low table entry is found, the following control gates are operated:

Low Load A3. Signal each low "find"

- A. Field control operates so that the transfer of the newly found lowest table entry from the Athr to A3 is controlled by the field control positions of the instruction.
- B. The program ring is recycled in preparation for the reset of A3 and the adding of the Athr contents to A3.

TLU Low R A3. Transfer new "low" argument to A3.

- A. [16] The Mem Latch 1 is turned on to signal that at least one low table value was found.
- B. [17] The contents of the Athr, under field control, are added to zeros inserted on channel 2. The Adder output is brought back to A3. Thus, the next lower argument enters A3.
- C. [15] The address of the lowest table factor found is parallel-transferred from Auxr 2 to positions 2 - 5 of A2.

TLU Low Incr RCW. Control TLU Ctrl Gate Con TLU Low.

- A. The program ring is recycled in preparation for incrementing of the RCW Start address.
- B. TLU Ctrl Gate C is turned on to cause incrementing and testing of RCW Start vs Stop addresses.

End TLU on Find.

- [19] If the Mem Latch 1 is on, RCW Sign is minus and No-Carry indicates Start is greater than Stop, TLU Control D is turned on to signal the end of the TLU Lowest operation.

TLU Control Gate D.

- [20] In addition to the usual functions of control gate D, the lowest found address is transferred from positions 2 - 5 of A2 to Auxr 2 so that it can be stored in the indexing portion of word 0098.

5.11.00 PRIORITY PROCESSING

An introduction to priority processing may be found in the 7070 Data Processing System Bulletin on Operation Codes, Form G24-7002, and in Section 5.0.00 of 7070 System Fundamentals (Volume I, Book A).

5.11.01 Priority Circuit Operation

Figure 5.11-1 summarizes the interrupt operation. The circled numbers on Figure 5.11-1 refer to the Card-Inquiry logic on Figure 5.11-2A and B and the sequence of operation (Card-Inq) on Figure 5.11-3. As indicated by ⑥ through ⑩ A on Figure 5.11-1, a different sequence is followed for tape-disk priority. The additional circuits required for tape-disk operations are illustrated by Figures 5.11-4 and 5.11-5. The Card-inquiry operation is discussed first.

Card-Inquiry Interrupt

- ① At the end of any card or inquiry operation a signal is sent to the 7602 to turn on a corresponding stacking latch. This latch remains on until just after the 7070 goes into the interrupt mode.
- ② When the interrupt ring reaches A1, B0, C0 and the Allow Card A Mask output is on, the Card A Stacking Latch output is tested to turn on both Card Inquiry Interrupt and Any Interrupt. The Card Inq Interrupt signal causes the sequence ① ② ③ ④ ⑤ ⑪ ⑫ ⑬ ⑭ ⑮ to take place.
- ③ The Any Interrupt Signal is switched with a WP on Logic 55.06.06 to turn on a timer latch. The preliminary latch is used to prevent any stray spikes from occurring on the Intr Waiting line that is sent to the A&P Controls. The latch output is then switched with a YP to turn on the Interrupt Waiting latch. Provided the Allow Intr signal (Mem Sync Channel Interlock) is on and the Interrupt Mode latch is off, the interrupt operation begins at the end of the data cycle.

This can happen either at the end of the Interrupt Release Op Code, provided another interrupt is waiting, or at the end of any instruction during a main program routine. If the main routine program step happens to end in a store in memory operation, the start of the interrupt is delayed until Store Control 5 time. Because this type of end data operation occurs only at Store Control 1 time, a delay latch can be turned on to postpone the start of the interrupt operation until Store Control 5 time.

- ④ With Intr Waiting, Not Intr Mode and End Data Operation signals on, Address 0097 control is turned on. As shown in Figure 5.11-2A, the address 0097 is generated, and a normal store-in-memory operation causes the next main routine instruction address to be transferred from the IC to core storage address 0097. At Store Control 4A time of the Addr 0097 Ctrl operation the Address 0100 Control latch is turned on.

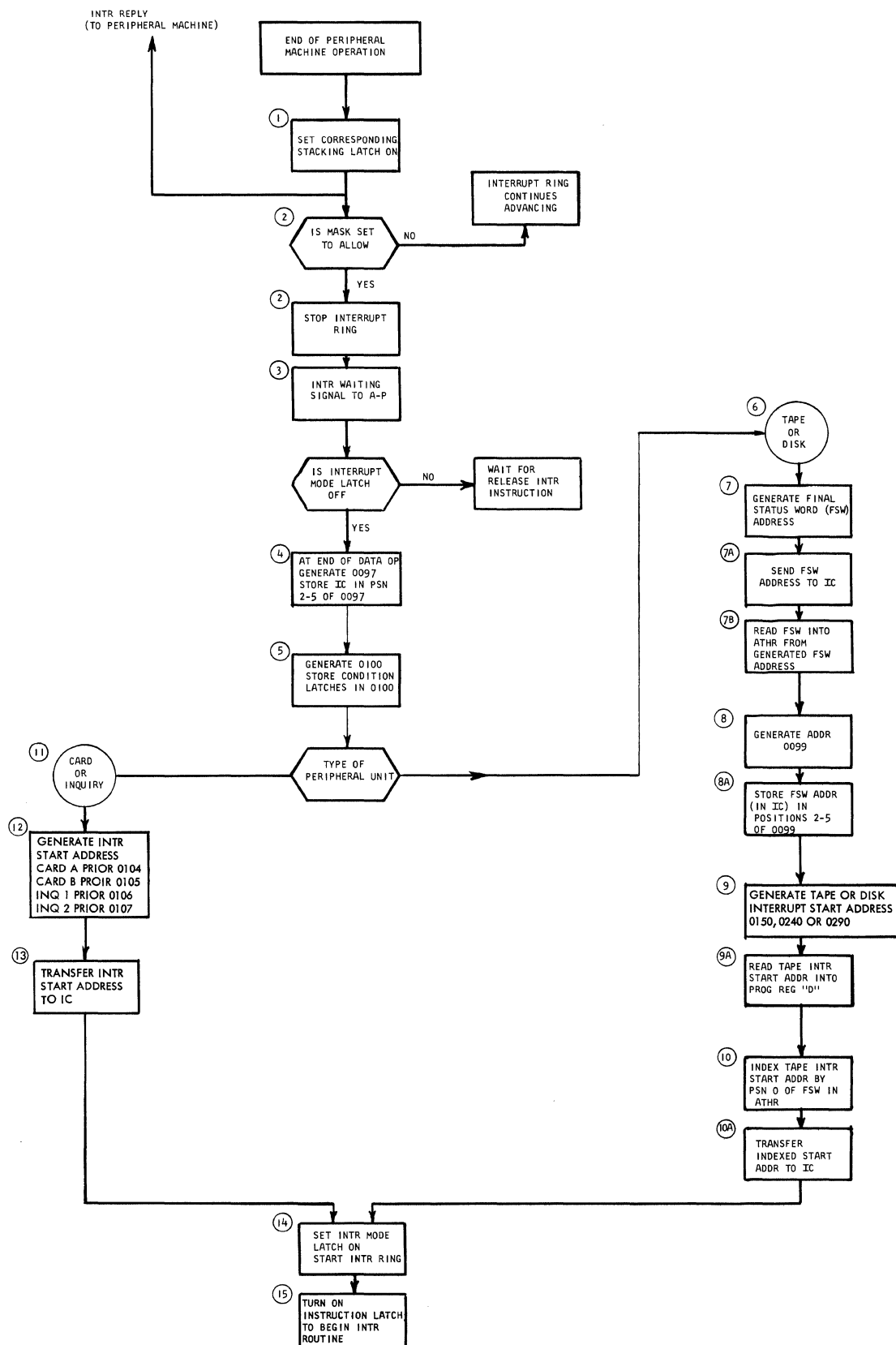


FIGURE 5.11-1. FLOW CHART FOR PRIORITY CIRCUIT OPERATION

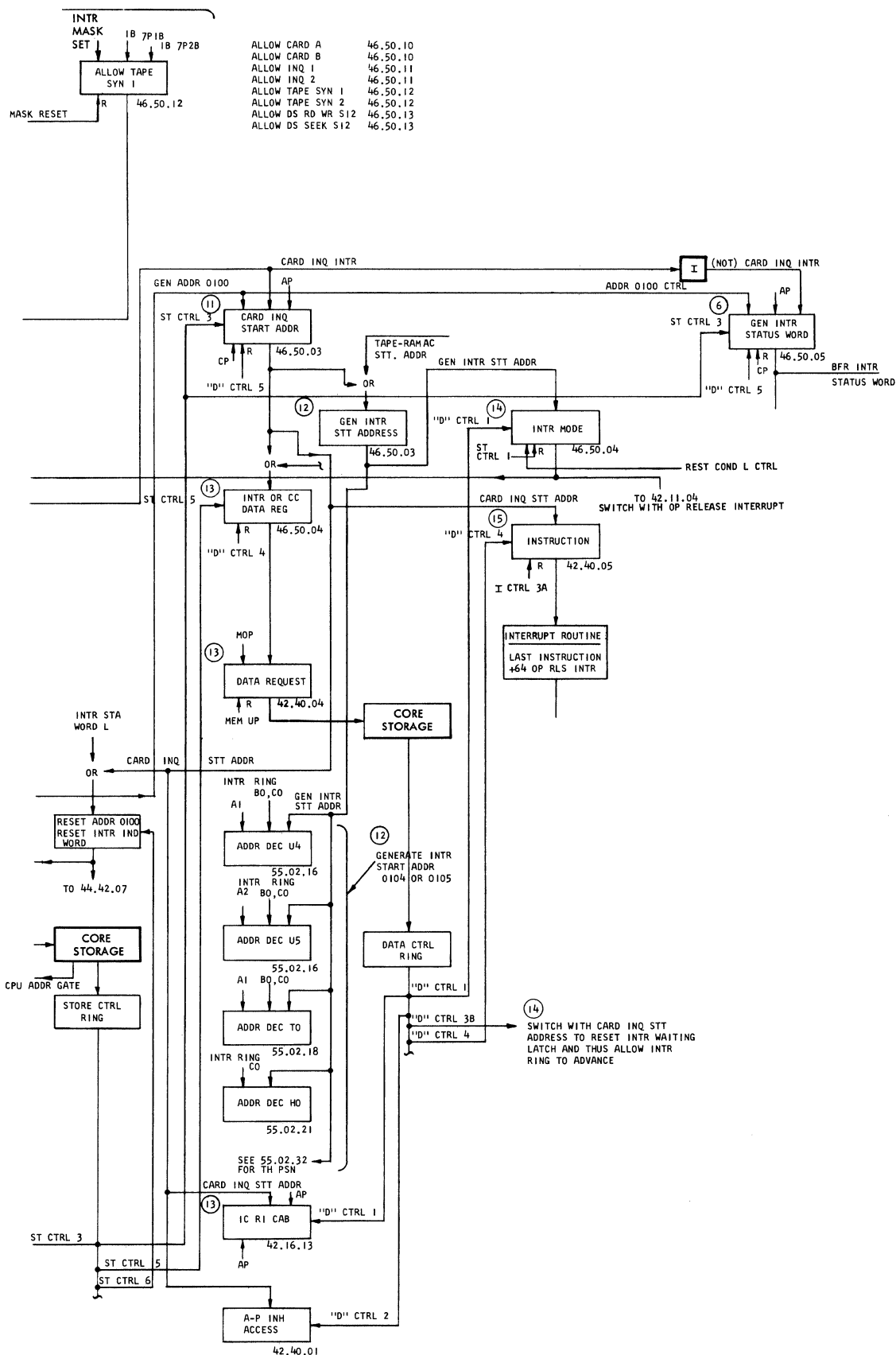


FIGURE 5.11-2B. LOGIC FOR INTERRUPT OPERATION (CARD OR INQ)

- ⑤ Address 0100 Ctrl initiates an operation very similar to ④. The address 0100 is generated and a store-in-memory operation transfers the setting of the condition latches via the IB to memory location 0100.
- ⑪ If a card or inquiry stacking latch initiated the operation, the Card Inquiry Start Address latch is turned on to control the generation of the particular interrupt start address to which the program will branch to enter the interrupt routine.
- ⑫ The address generated is controlled by the position at which the interrupt ring stops. Figure 5.11-6 tabulates the interrupt start addresses for each type of peripheral unit. For example, if the Inquiry 2 Stacking Latch had been turned on, the interrupt ring would be stopped at A4, B0, C0 and an Interrupt Start Address of 0107 would be generated.
- ⑬ A data-request operation causes the transfer of the generated Intr Start Address from the Address Generator via the CAB to the Instruction Counter.
- ⑭ The interrupt-mode latch is turned on at D ctrl 1 time. The output of Intr Mode is inverted to prevent switch 3 (Figure 5.11-2) from starting another interrupt operation. When the Intr Mode latch is turned off by the release-interrupt instruction, another Intr Waiting signal can be recognized.

At D Ctrl 3B time, the Intr Waiting latch is turned off. This allows the Intr Ring to begin advancing again.

- ⑮ The instruction latch is turned on, and the first program step of the interrupt routine is taken from the address placed in the IC during step ⑬.

The last instruction in the interrupt routine must be Op + 64, Priority Release so that the interrupt-mode latch may be turned off. Either of the following then occurs:

- a. If, during the interrupt routine, the interrupt ring has advanced to a point where another stacking latch is found On, another interrupt operation begins immediately with the turning on of Addr 0097 Ctrl.
- b. If no other stacking latches are on when the Intr Mode latch goes off, the priority-release instruction returns the 7070 to the main routine.

Tape-Disk Interrupt

Steps ① through ⑤ of a Tape-Disk Interrupt operate in the same general manner as Card-Inquiry Interrupt except that the Card-Inq Intr latch does not come on. Thus, Card-Inq Start Address ⑪ can not come on. Instead, Address 0100 Ctrl switches with St Ctrl 3 and Not Card-Inq Intr to turn on Generate Interrupt Status Word ⑥ on Figure 5.11-4). Recall that the following must be accomplished before the interrupt program routine can begin:

- a. The address of the final status word, developed at the end of the tape or disk operation, is stored in positions 2 - 5 of core storage location 0099. The final status word is read out from the FSW generated address to the arithmetic register in preparation for indexing the interrupt-start address.

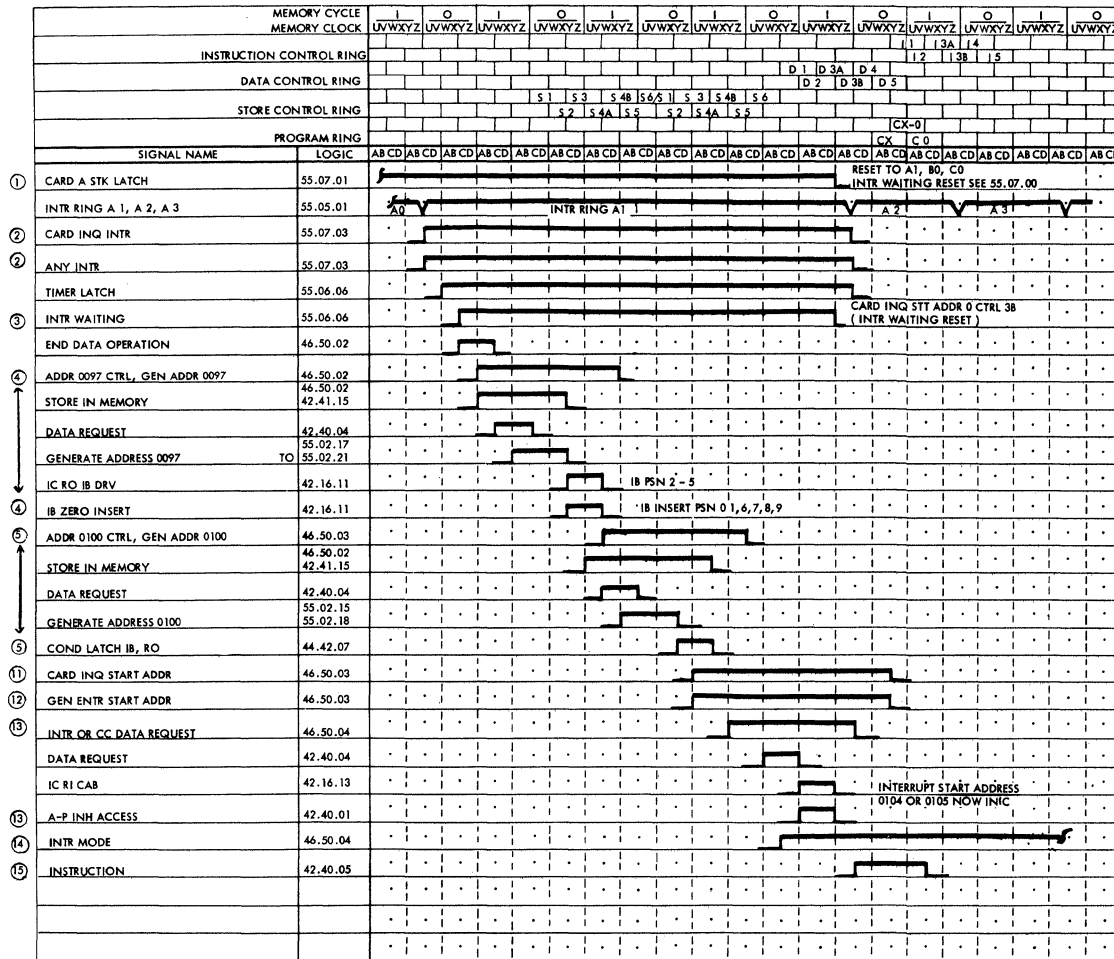


FIGURE 5.11-3. SEQUENCE OF INTERRUPT OPERATION (CARD UNIT)

FIGURE 5.11-6. INTERRUPT CORE STORAGE LOCATIONS

INTER RING POSITIONS	HUNDREDS	C 0								C 1							
	TENS	B 0	B 1	B 2	B 3	B 4				B 0	B 1	B 2	B 3	B 4			
	UNITS	CARD/INQ/TAPE								RAMAC							
		CARD/INQ	SYNC 1		SYNC 2		SYNC 3		SYNC 4		ACC 0		ACC 1		ACC 2		
		START ADDRESS	FINAL	INITIAL	FINAL	INITIAL	FINAL	INITIAL	FINAL	INITIAL	FINAL	INITIAL	FINAL	INITIAL	FINAL	INITIAL	
	A 0	ANY STK L	0110	0160	0120	0170	0130	0180	0140	0190	0200	0250	0210	0260	0220	0270	
		ON TEST	TD 0		TD 0		TD 0		TD 0		BOX 0						
	A 1	0104	0111	0161	0121	0171	0131	0181	0141	0191	0201	0251	0211	0261	0221	0271	
		CARD A	TD 1		TD 1		TD 1		TD 1		BOX 1						
	A 2	0105	0112	0162	0122	0172	0132	0182	0142	0192	0202	0252	0212	0262	0222	0272	
		CARD B	TD 2		TD 2		TD 2		TD 2		BOX 2						
	A 3	0106	0113	0163	0123	0173	0133	0183	0143	0193	0203	0253	0213	0263	0223	0273	
		INQ 1	TD 3		TD 3		TD 3		TD 3		BOX 3						
	A 4	0107	0114	0164	0124	0174	0134	0184	0144	0194							
		INQ 2	TD 4		TD 4		TD 4		TD 4								
	A 5		0115	0165	0125	0175	0135	0185	0145	0195							
			TD 5		TD 5		TD 5		TD 5								
	A 6		0116	0166	0126	0176	0136	0186	0146	0196							
			TD 6		TD 6		TD 6		TD 6								
	A 7		0117	0167	0127	0177	0137	0187	0147	0197							
			TD 7		TD 7		TD 7		TD 7								
	A 8		0118	0168	0128	0178	0138	0188	0148	0198							
			TD 8		TD 8		TD 8		TD 8								
	A 9		0119	0169	0129	0179	0139	0189	0149	0199							
			TD 9		TD 9		TD 9		TD 9								

INTERRUPT START LOCATIONS

CARD A	0104	
CARD B	0105	
INQ 1	0106	
INQ 2	0107	
TAPE	015X	INDEXED BY PSN 0 OF FINAL STATUS WORD
RAMAC R/W	024X	
RAMAC SEEK	029X	

RESERVED LOCATIONS FOR INTERRUPT

INTERRUPT REGISTER (MAIN ROUTINE START ADDRESS)	0097
FINAL STATUS WORD ADDRESS REGISTER	0099
CONDITION LATCH REGISTER	0100

- b. The interrupt-start address is generated and then indexed by adding the contents of the digit zero position of the FSW (in the Athr) to the units position of the generated Intr Start Address.
- ⑦ A different final-status-word address is generated, depending upon which tape or disk stacking latch initiated the operation. Figure 5.11-6 shows the initial and final status word addresses for all tape and disk units. For example, if Sync 3 Tape Drive 4 turned on the stacking latch, the C0, B3 and A4 Intr Ring positions would be on. According to Figure 5.11-6 the generated FSW address would be 0134.
 - ⑦A A data request operation sends the generated FSW address to core storage and to the IC at D Ctrl 1 time.
 - ⑦B The final status word is next transferred from the addressed core storage location to the arithmetic register via IB in the usual manner. The digit 0 position of the FSW is used later to index the interrupt start address.
 - ⑧ The address 0099 is generated so that the FSW address may be stored in positions 2 - 5 of core storage location 0099.
 - ⑧A A store-in-memory operation causes the FSW address in the IC (Connected to the IB Positions 2 - 5) to be stored in positions 2 - 5 of location 0099. Zeros are inserted on the IB in positions 0, 1, 6, 7, 8, 9.
 - ⑨ The basic tape or RAMAC interrupt start address is generated. The stopping point of the interrupt ring determines which address is generated. Figure 5.11-6 indicates the interrupt start addresses.
 - ⑨A During a data-request operation (access inhibited) the generated start address is read into program register D via CAB.
 - ⑩ The start address is indexed by adding the digit 0 position of the Athr to the start address in program register D.

Basic Start Address	0 1 5 0
Athr Psn 0 = 3	+ <u>0 0 0 3</u>
Actual Start Address	0 1 5 3

- ⑩A The indexed start address is transferred to the IC in preparation for the beginning of the interrupt program routine.
- ⑭ The interrupt-mode latch is turned on at D Ctrl 1 time. The output of Intr Mode is inverted to prevent switch 3 (Figure 5.11-2) from starting another interrupt operation. When the Intr Mode latch is turned Off by the priority release instruction, another Intr Waiting signal can be recognized.
- ⑮ The instruction latch is turned on, and the first program step of the interrupt routine is taken from the address placed in the IC during step ⑩A.

Stacking Latch Set, Op -61, and Reset, Op -62

Introduction

Op Code -61 turns on, and Op Code -62 turns off the stacking latch specified by positions 4 and 5 of the instruction. Positions 4 and 5 designated the stacking latches to be set or reset in accordance with the following:

01, 02	Unit Record Latch A, B
03, 04	Inquiry Controls 1, 2
10-19	Tape Units 0-0 on Channel 1
20-29	Tape Units 0-9 on Channel 2
30-39	Tape Units 0-9 on Channel 3
40-49	Tape Units 0-9 on Channel 4
50-53	Access arm 0, Disk Storage Unit 0-3
60-63	Access arm 1, Disk Storage Unit 0-3
70-73	Access arm 2, Disk Storage Unit 0-3

Circuits for Stacking Latch Set. Figure 5.11-7 shows the logic for stacking latch set. This is a No- Access Code so the program ring is immediately recycled. The Op Code, Op STLS -61, is developed in the usual manner. This signal switches at CX time with the field register tens position, decimal 1, output and with the field register units position, decimal output to turn on the Syn 1, TD0 Stacking Latch. The Op STLS -61 signal causes a Set Test Reset signal which in turn switches with the field register tens position, decimal 1 output to cause a Test Code Syn 1 signal. This signal is now switched with the field register units position decimal 0 output and with the output of the newly "set" latch to turn on the test latch.

A check is now made to be certain the test latch comes on when Op STLS is programmed. A Tran to D signal is developed, but the No Tran to D signal is not developed. Thus Branch to D does not come on if the designated stacking latch is properly set. If the stacking latch is not set, Br to D comes on, switches with No Branch (forced on) to cause an error stop. A Restart A&P is automatically signalled at C1 time to continue the machine's operations. A Branch-No Branch error stops the machine, regardless of the Restart.

Circuits for Stacking Latch Reset. Figures 5.11-8, and 9 show the circuit logic and sequence of operation for stacking latch reset. This operation is similar to stacking latch set except that the reset code signal switches with the field register units position decimal output to turn off the designated stacking latch. A check is made to be certain that the stacking latch is turned off as shown in Figure 5.11-8. If the designated latch is not turned off, the test latch comes on and causes the generation of a "Transfer to D" signal and thus causing a Branch-No Branch error in the A&P unit.

The computer is automatically restarted with the commencement of this operation.

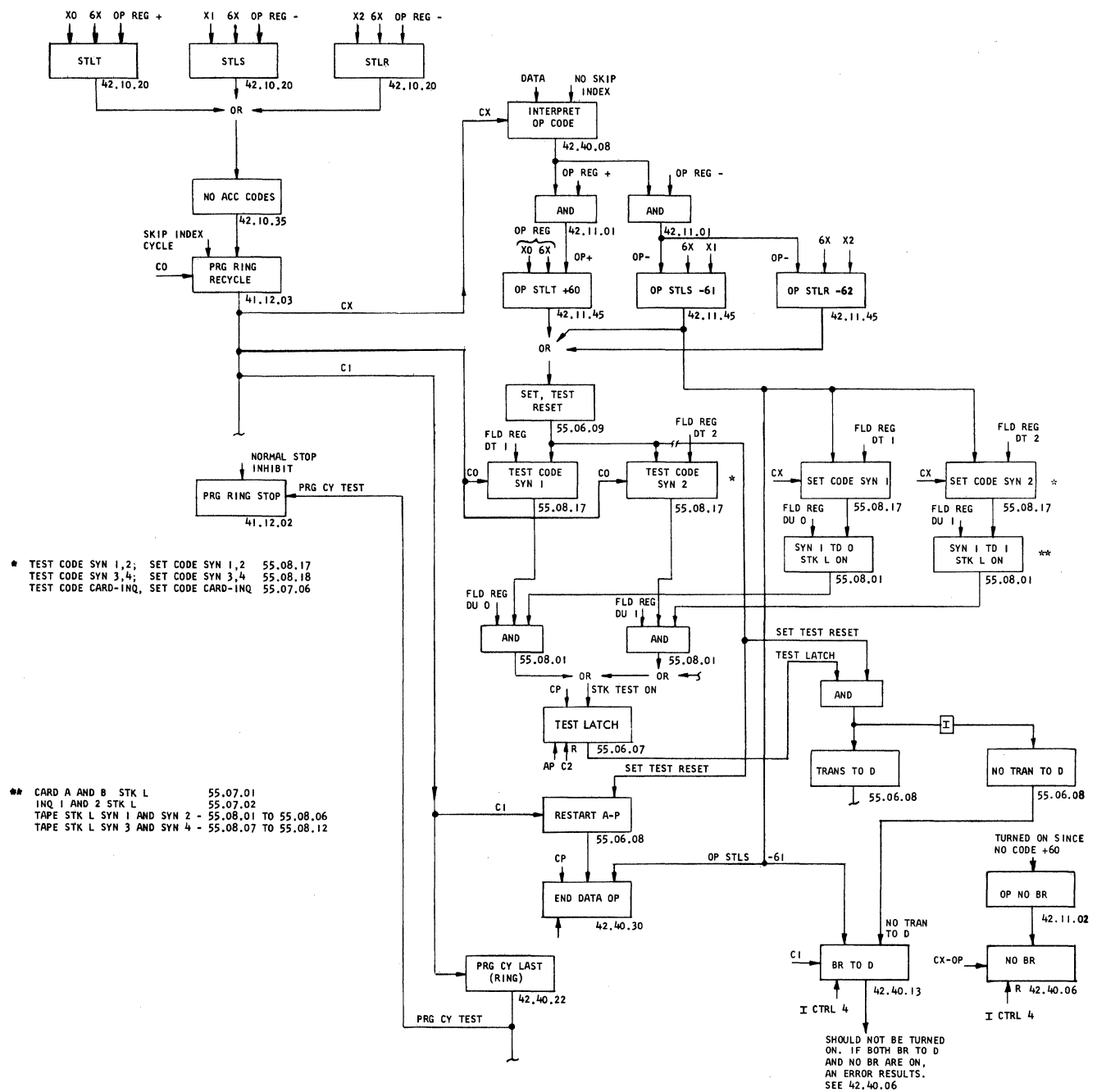


FIGURE 5.11-7. LOGIC FOR STACKING LATCH, OP-61

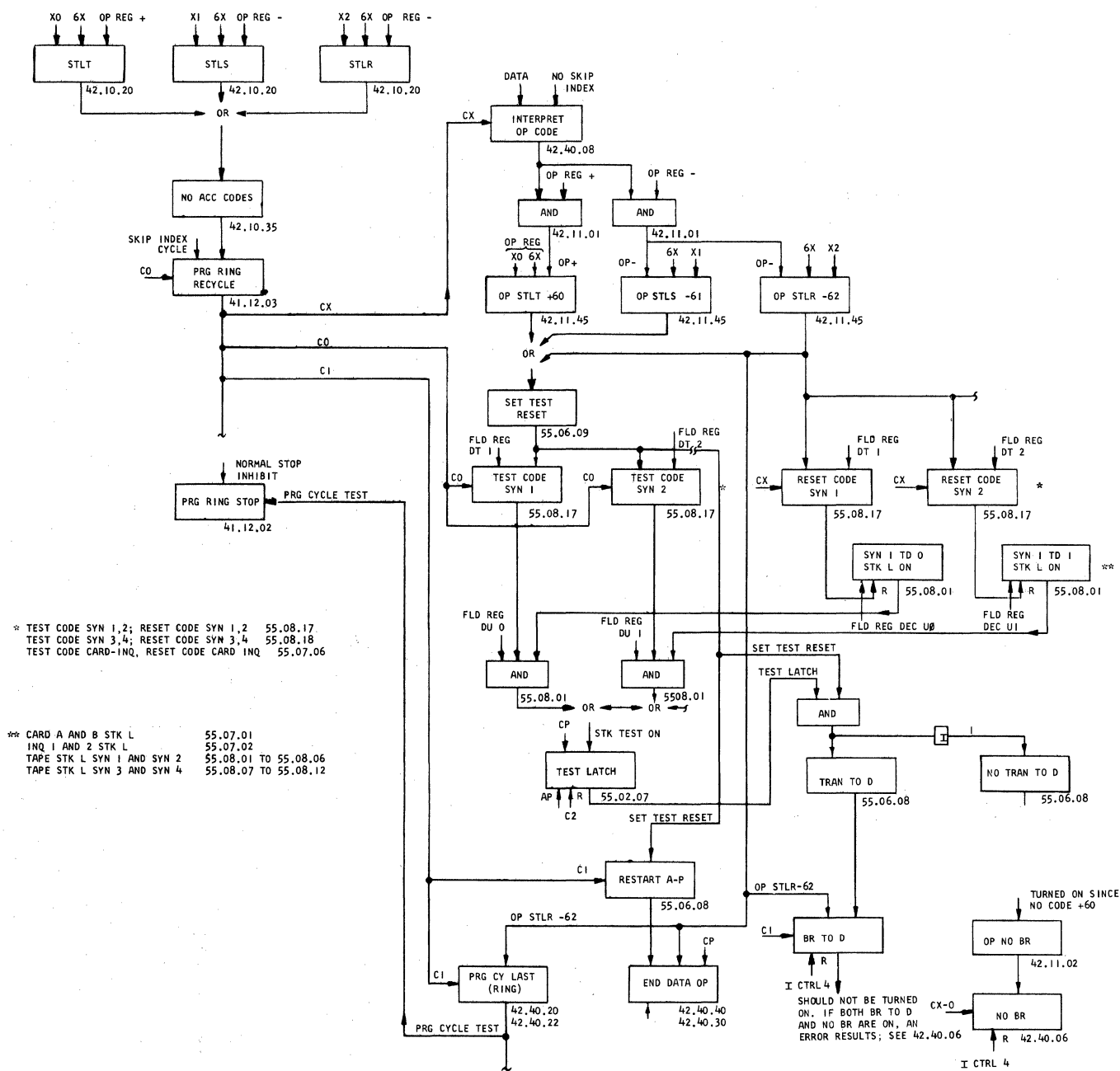


FIGURE 5.11-8. LOGIC FOR STACKING LATCH RESET, OP-62

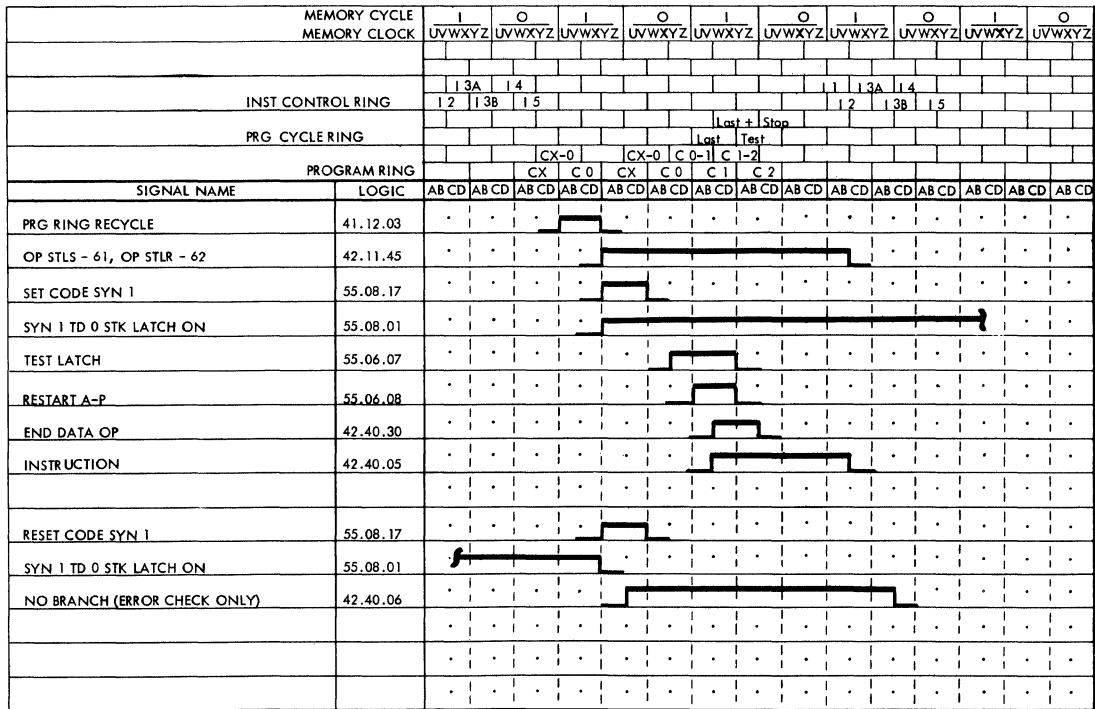
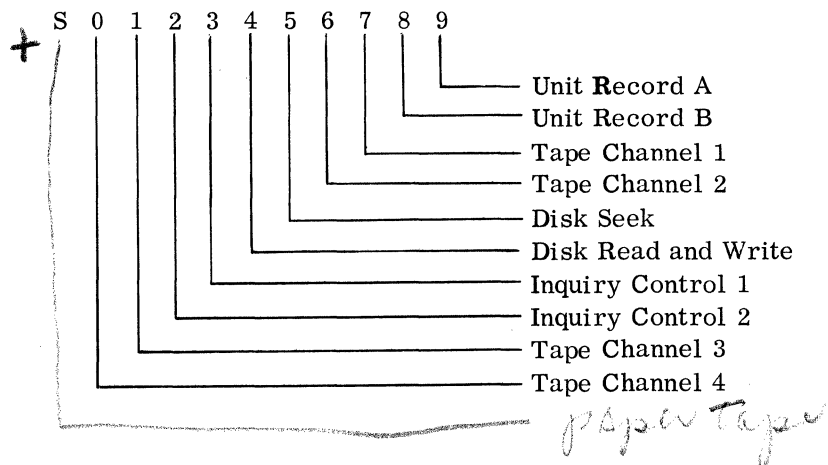


FIGURE 5.11-9. SEQUENCE FOR STACKING LATCH SET AND RESET OP CODES -61 AND -62

5.11.02 Priority Codes

Priority Control, Op + 55 (PC)

Introduction. This operation sets the priority masks. The word in the storage location designated by program register D is used to set the masks. A zero in a digit position of the addressed word allows priority and a one digit prohibits priority. The digit positions control the setting of the masks as follows:



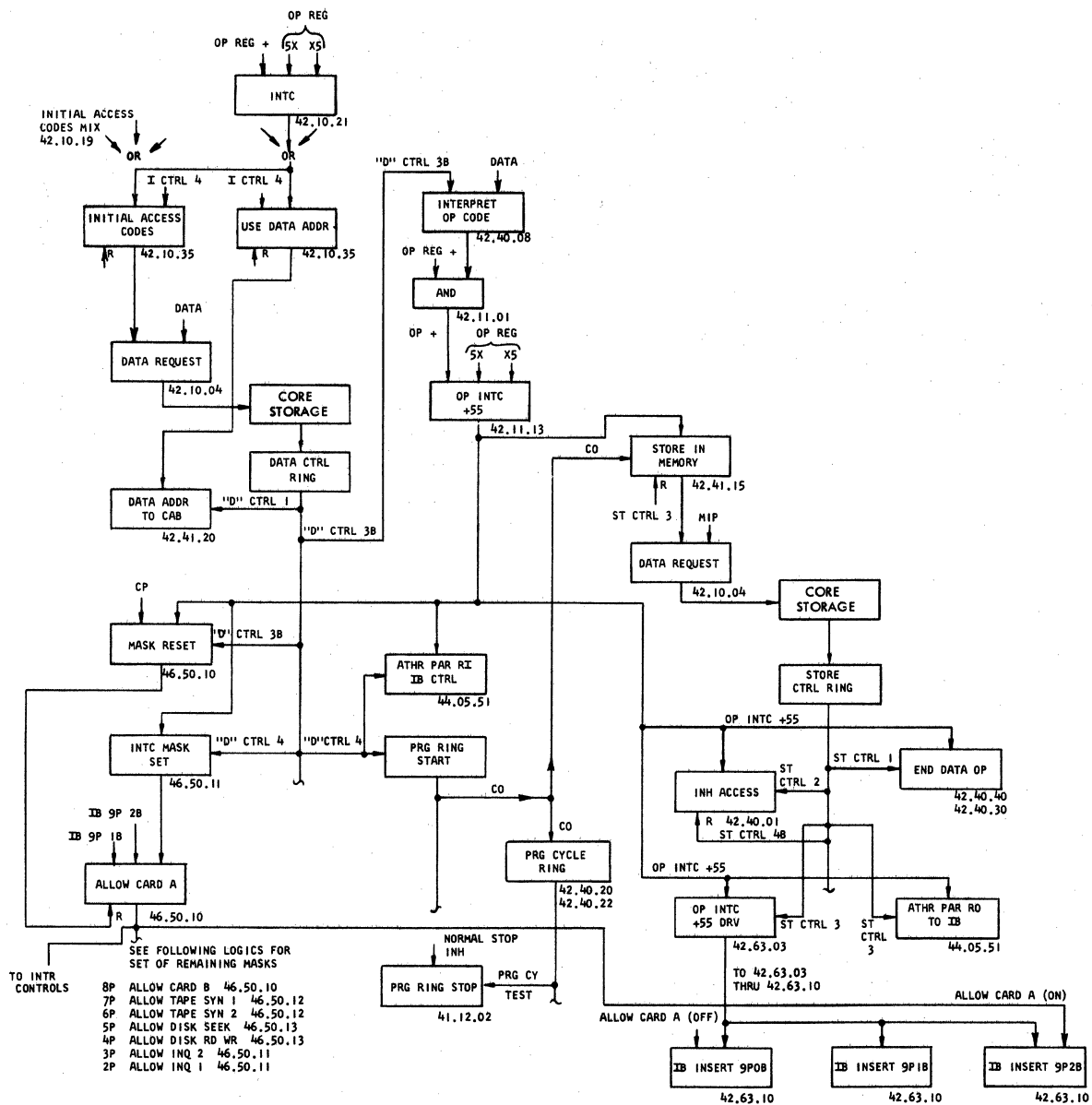


FIGURE 5.11-10. LOGIC FOR PRIORITY CONTROL, OP +55

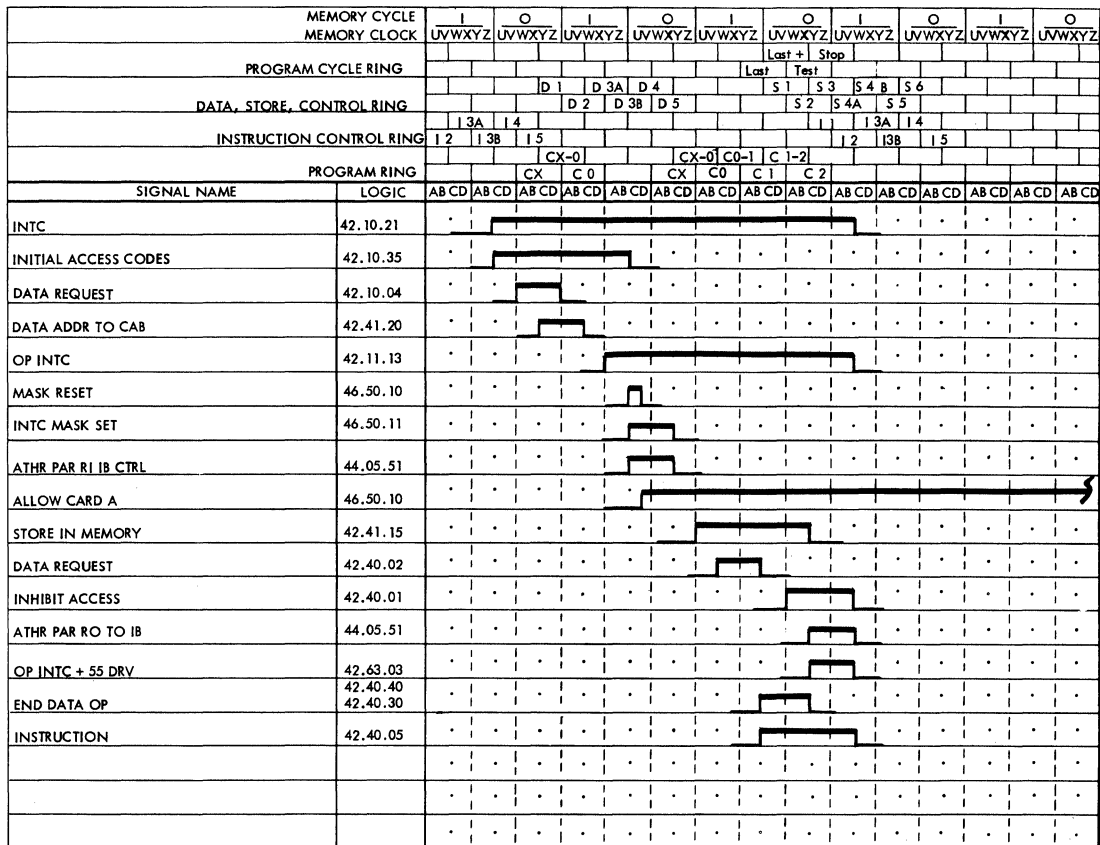


FIGURE 5.11-11. SEQUENCE FOR PRIORITY CONTROL, OP +55

The masks, as set by the priority control instruction, are effective until they are changed by the next priority control instruction or computer reset. Positions 4 and 5 of the instruction word are not used.

Circuits for Priority Control. Figure 5.11-10 shows how a memory access operation causes the mask word specified by the address in program register D to read out to the A&P unit. At D Ctrl 3B, CP time all masks are reset; at D Ctrl 4 time, all Allow latches are set if the associated digit position contains a digit 0. Also at D Ctrl 4 time, the mask word is read into the arithmetic register. The Athr contents are later checked against the Allow latch outputs to be certain the latches are set correctly. This is accomplished by the information-bus validity-check circuits. A store-in-memory operation, access inhibited, controls the checking operation as shown on Figure 5.11-7. The mask is read out to the IB from the Athr. If the 9th position of the arithmetic register contains a digit 0, the 1 and 2 bit IB capacitors are charged. If the Allow Card A latch is on, 1 and 2 bits are inserted on the information bus. Thus, only two bits occur on the IB in the 9th position and no validity-check error results. However, if the Allow Card A latch was not set, 0 and 1 bits would be inserted on the IB and a validity error would occur with 1 and 2 bit capacitors charged. Figure 5.11-11 indicates the sequence of operation.

Stacking Latch Test, Op + 60

Introduction. This operation tests the stacking latch designated by positions 4 and 5 of the instruction. If the latch is on, the next instruction is taken from the address

in program register D; if it is off, from the instruction counter. The field register, positions 4 and 5 of the instruction, specify the stacking latch to be tested as follows:

00	Any Stacking Latch
01, 02	Unit Record Latch A, B
03, 04	Inquiry Controls 1, 2
10-19	Tape Units 0-9 on Channel 1
20-29	Tape Units 0-9 on Channel 2
30-39	Tape Units 0-9 on Channel 3
40-49	Tape Units 0-9 on Channel 4
50-53	Access Arm 0, Disk Storage Unit 0-3
60-63	Access Arm 1, Disk Storage Unit 0-3
70-73	Access Arm 2, Disk Storage Unit 0-3

Circuits for Stacking Latch Test. Figures 5.11-12, 13 show the circuit logic and sequence of operation for Stacking Latch Test. The Program Ring Recycle, development of the Op code, and the Set, Test, Reset signal occur in the same manner as with Op Stacking Latch Set. A test-code signal is developed at C0 time from the field register tens position decimal output. The test signal is then switched with the field-register-units position output and the selected Stacking Latch On to cause a Stacking Test On signal. Thus, if the selected latch is on, (i. e., selected according to the preceding table) the test latch comes on and a Transfer-to-D signal results in Branch to D in the same manner as with any other branch code. End Data Operation is signalled only if either No Branch or Branch to D comes on. The usual branch-no branch check is made because No Br is not turned on automatically as with no branch codes. See Figure 5.11-12 for the manner in which No Br is kept off and logic 42.40.06 for the Branch-No Branch check circuit.

When the field register contains 00, any stacking latch on causes a Branch to D. This is accomplished as shown on Figure 5.11-12. Field Register DT 0 turns on Test Code Card Inq. This in turn switches with Any Stacking Latch On and Field Register DU 0 to turn on the Test Latch.

Priority Release, Op + 64

Introduction. This code is usually the last instruction in a priority program routine. It releases priority so that the main program can continue (Unless another priority is waiting). There are four courses of action which can be taken depending upon whether another priority is waiting and whether the data address of the priority release instruction is 0097 or not. Recall that the next main routine address was stored in positions 2 - 5 of 0097. The four courses of action are illustrated in Figure 5.11-14 and summarized below.

1. No Priority Routine waiting, address in PR D is 0097.

The address in positions 2 - 5 of word 0097 is sent to the IC and the next main routine instruction is taken from the IC address.

2. No Priority Routine waiting, Address is not 0097.

The address in PRD is transferred to the IC and the main routine starts with the instruction in the IC address. Word 0097 is unchanged.

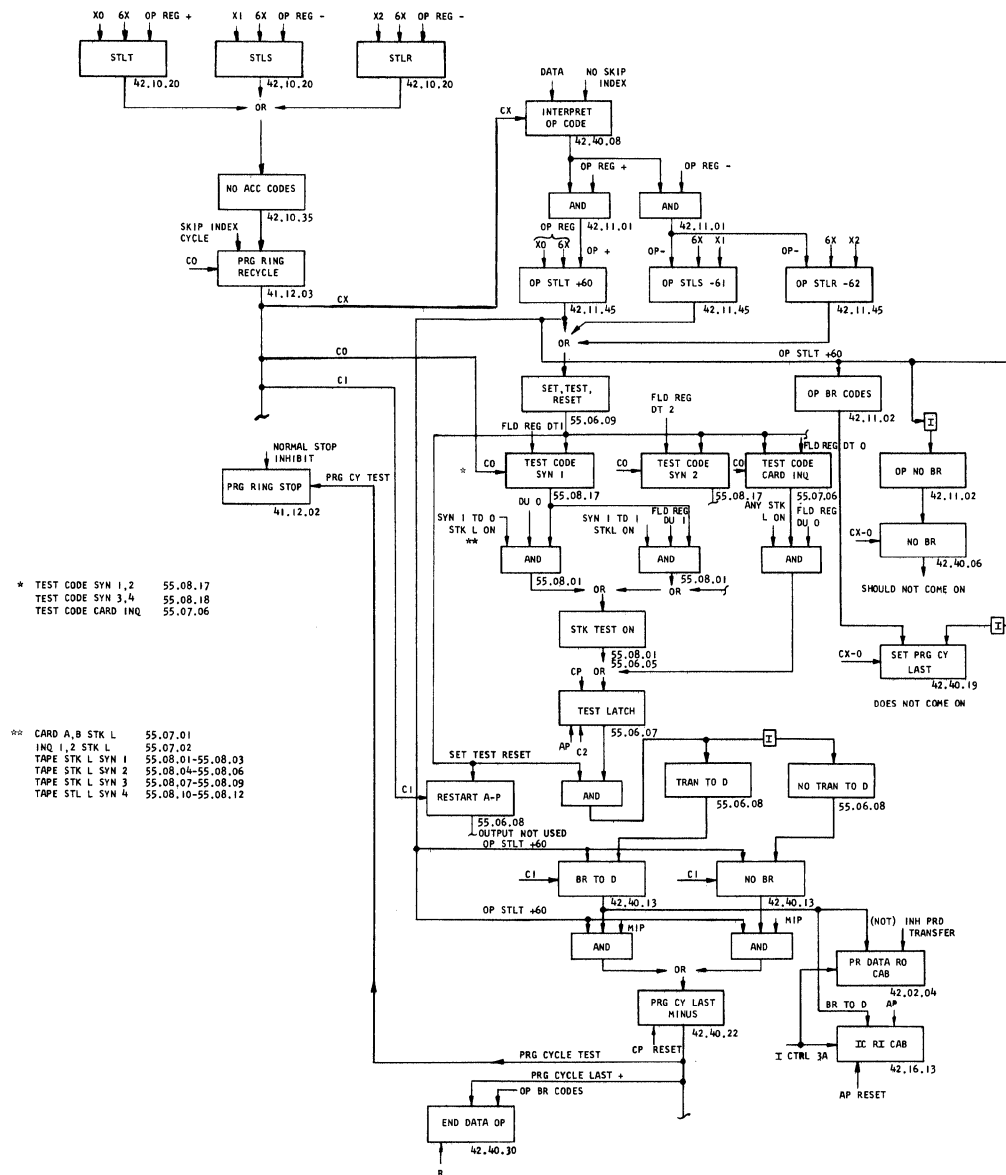


FIGURE 5.11-12. LOGIC FOR STACKING LATCH TEST, OP +60

3. Another priority routine waiting, address is 0097.

The waiting priority routine is started. Word 0097 is unchanged by the new priority routine.

4. Another priority routine waiting, address is not 0097.

The waiting routine is started. From 2 above, note that when the PR D address of the Release Intr instruction is not 0097, positions 6 - 9 of the PR is transferred to the IC. When the 2nd Interrupt begins the IC is stored in positions 2 - 5 of word 0097. Thus, the PR D address is placed in positions 2 - 5 of word 0097 and the return address to the main routine is lost, unless it was stored elsewhere during the 1st priority routine.

When the priority release command is executed with the machine in the non-interrupt mode, a No-Op is performed and the program proceeds to the next location in sequence.

Priority Release Circuits. Circuit Logic for priority release is shown in Figure 5.11-15. See Figure 5.11-16 for the sequence of operation. Circled numbers, tie together like functions on the various priority release illustrations.

- ① Op + 64 is presensed and switched with Intr Mode. Whenever the machine has been in a interrupt mode, the release interrupt functions are performed.
- ② If Op + 64 is programmed when the machine is not in an interrupt mode, a No Op is signalled.

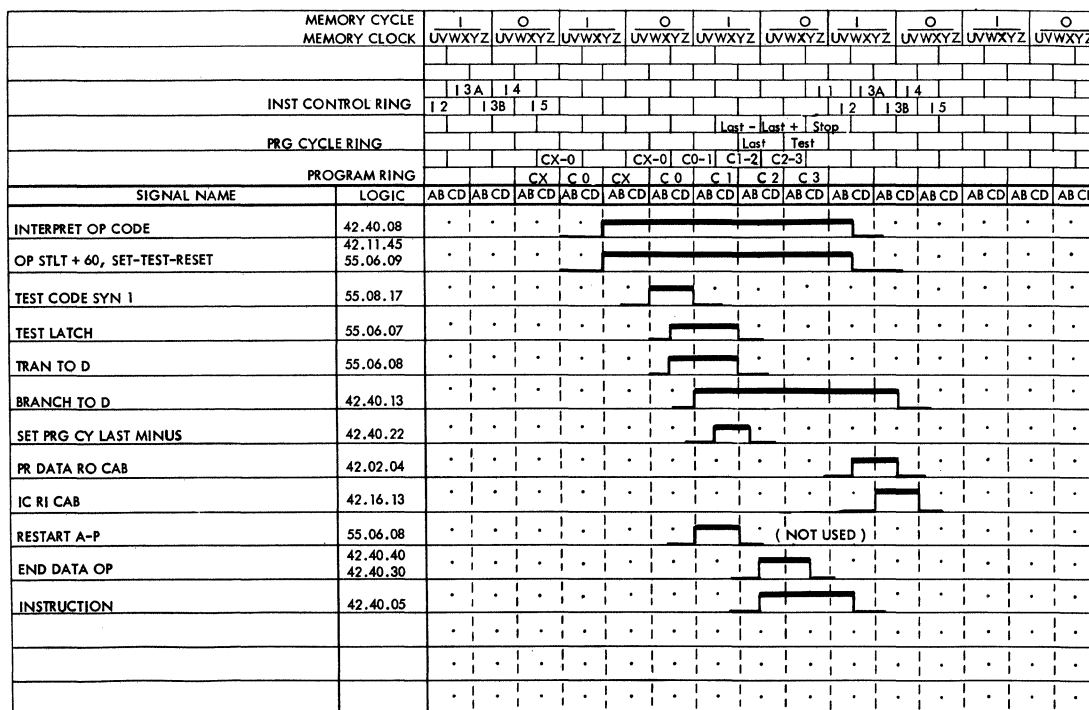


FIGURE 5.11-13. SEQUENCE FOR STACKING LATCH TEST, OP +60
(NO INDEXING)

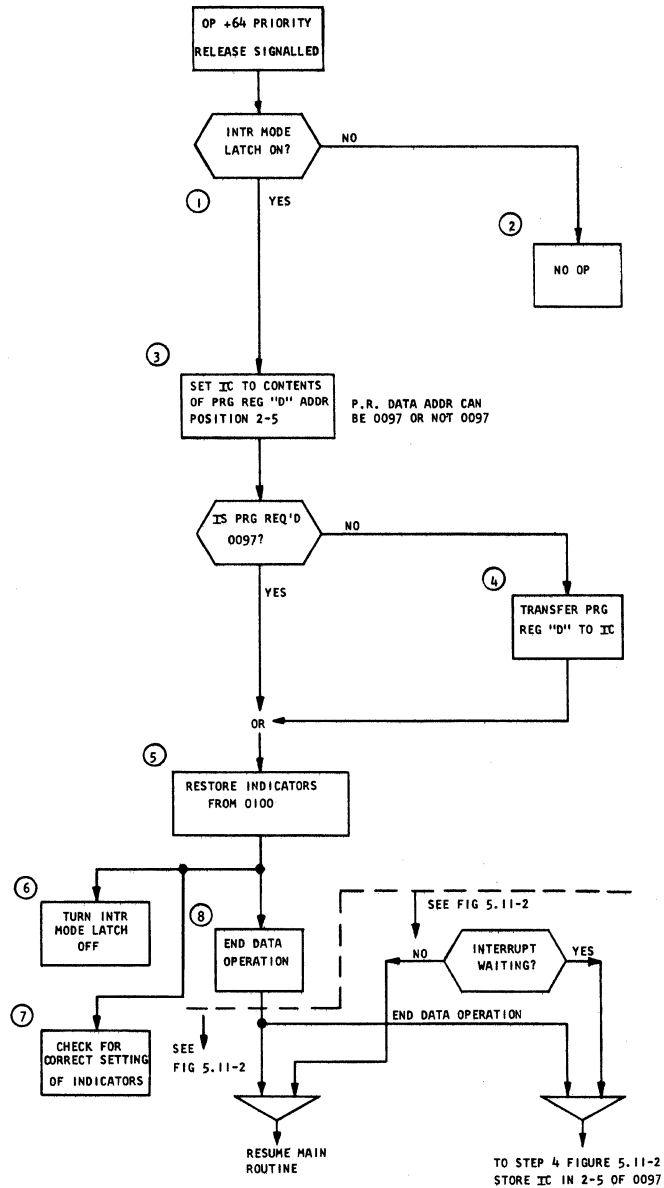


FIGURE 5.11-14. FLOW CHART FOR PRIORITY RELEASE, OP +64

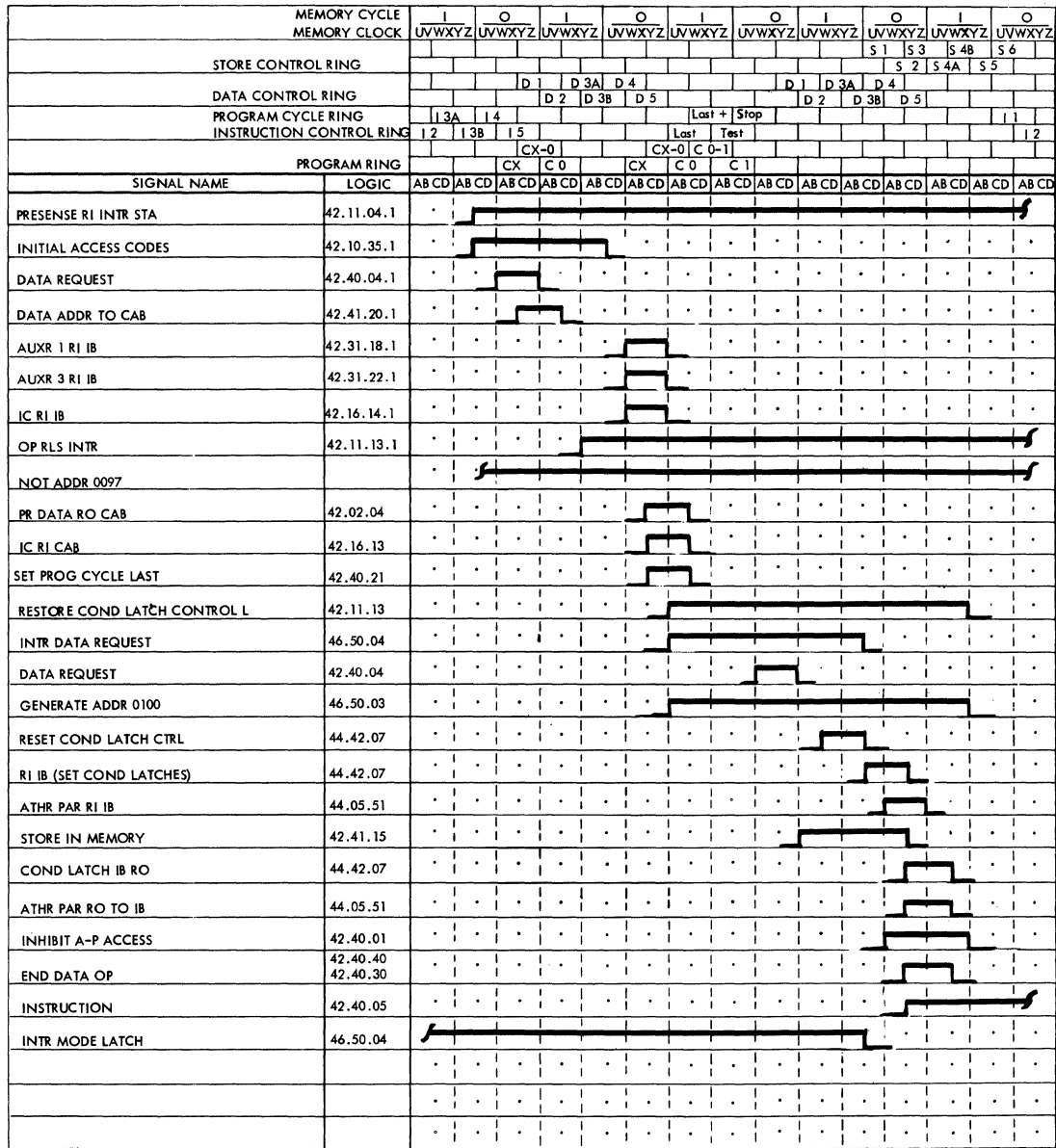


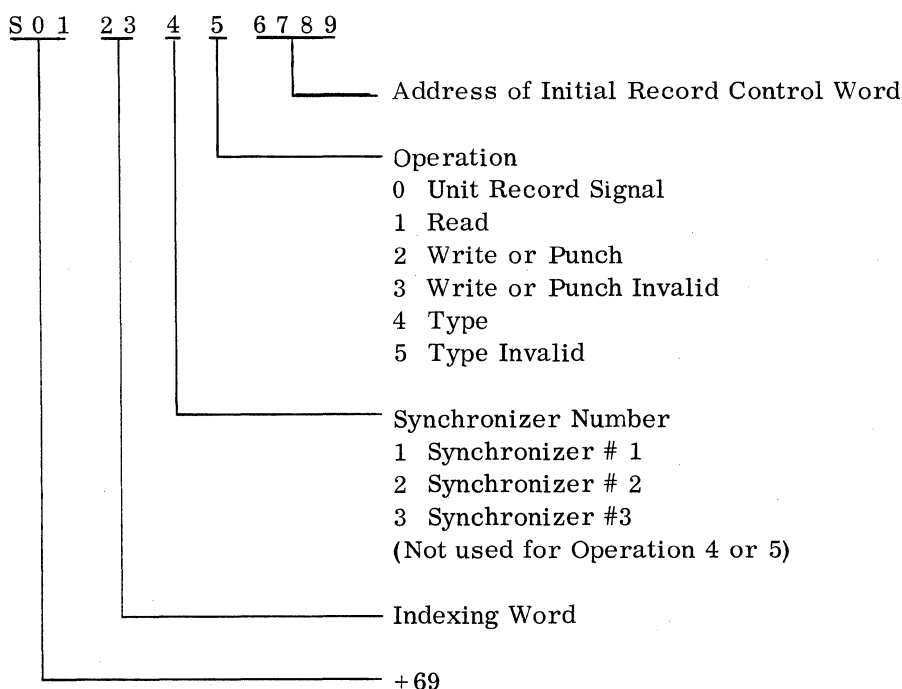
FIGURE 5.11-16. SEQUENCE FOR PRIORITY RELEASE, OP +64

- ③ An initial access data request operation brings the contents of positions 2 - 5 of the core storage location specified by the address in program register D to the instruction counter. The instruction counter now contains the return to the main routine if PR D contained 0097. If the contents of PR D is not 0097, whatever is read into the IC by the data request will be replaced by positions 6 - 9 of the instruction (Example 1 2 3 4) during step 4.
- ④ The program ring is started at Data Ctrl 4 and the CX-0 output of the ring gates the transfer of the not 0097 address from PR D to the IC. This happens only if the PR D does not contain 0097.
- ⑤ A data request brings the contents of location 0100 to the condition latches. The address 0100 is generated.
- ⑥ The Interrupt Mode latch is reset at Store Control 1 time.
- ⑦ During the data request operation of ⑤, the condition latch setting is read into the arithmetic register as well as into the condition latches. A store-in-memory operation is started, with memory access inhibited, to time a comparison of the condition latch settings against the arithmetic register. At Store Control 3 time both condition latches and arithmetic register are read out to the information bus. If the bits from the two sources do not agree, extra bits will be detected by the IB Validity Check circuits.
- ⑧ End Data Operation is signalled at Store Control 4B time. If another interrupt is waiting, the output of the switch labelled ③ on Figure 5.11-2 switches with End Data Op to turn on Intr Start and Addr 0097 Control to begin another interrupt cycle. Note that the output of the switch ③, Figure 5.11-2, also inhibits the turn on of the instruction latch. Thus, End Data Op is prevented from a new instruction cycle if an interrupt is waiting.

5.12.00 CARD CONTROL OPERATIONS

All operations in the IBM 7500 Card Reader, 7400 Printer and the 7550 Card Punch are controlled by the + 69 Card Control Op Code. In addition, the typewriter on the IBM 7150 Console may be used as an output device under control of Op Code + 69. On a read operation, data are moved from input synchronizer to core storage. On punch, print, or type operations, data are moved from core storage to an output synchronizer. The machine connected with the particular synchronizer is then operated for one cycle. All movement of data between core storage and input-output synchronizer is under control of record control words.

The format of the card control instruction is as follows:



The various operations will be described in the following sections. See 5.5.01 for a description of the use of Record Control Words.

5.12.01 Unit Record Signal

Introduction

A zero in position 5 of the instruction conditions the card reader so that an impulse will be emitted from the program exit switch control panel hub just before 9 CB time of the next reader cycle. The impulse can be used to pick up selectors, impulse the off-set stacker, etc. The IC is advanced by one. The code is not used with output.

Circuits

Figure 5.12-1 shows the logic for the setting of the Buffer 1 Program Exit Switch. The operation begins with the normal data request operation to read out the RCW from core storage. The RCW is not used in the operation; however, for circuit simplicity it is allowed to enter the auxiliary register in the usual manner. When the field register units position contains zero, an Op 0 signal is developed. This is sent to the 7600 along with a Buffer 1 signal to set the Program Switch 1 latch. As soon as the latch is set, a restart signal is returned to the A-P circuits to start end-data operation. The instruction latch is then turned on and the program continues. The program exit switch in the reader is set when the next card control read operation calls for Buffer 1.

5.12.02 Read Operation

Introduction

This operation causes a block transmission of up to 16 words from the input synchronizer designated by the D4 position of the instruction to the core storage area specified by the record control words.

The address of the first RCW is specified by positions 6, 7, 8, 9 of the instruction. The start address of the first RCW specifies the core storage location in which the first word from the drum buffer is stored. All 16 words are moved from the buffer under control of the record control words unless the stop address of the last RCW is encountered first. In this case the last word is transmitted to the stop address location of the last RCW.

Figure 5.12-2 shows the functions performed during each cycle of the program ring for a specific example of a read operation. After the card control operation, the pre-sensing of data request operation takes place as with any initial access Op code. This causes the RCW in the core storage location indicated by the address in PRD be brought to the auxiliary register ①.

- ② The first RCW address is next transferred to the Address Control Register (CR). After each use of a new RCW, the address in the CR must be increased by 1 by the one-upper circuit in preparation for reading out the next RCW.
- ③ The Card Control Op code is analyzed. On the basis of a 1 in the D4 position of the instruction, a Bfr 1 signal is sent to the input-output synchronizer. On the basis of a 1 in the D5 position of the instruction, a Op 1 (Read) signal is sent to the synchronizer.
- ④ When the drum has turned to the beginning of the Buffer 1 area, a DX Start Transfer signal is returned to the A-P circuits to start the serial transfer of data from the drum to the word buffer register. The DX Start transfer signal also causes the program ring to start. The ring runs in synchronism with the turning of the drum. The program ring is recycled at each C10 time until after the last data are transferred from drum to WBR.

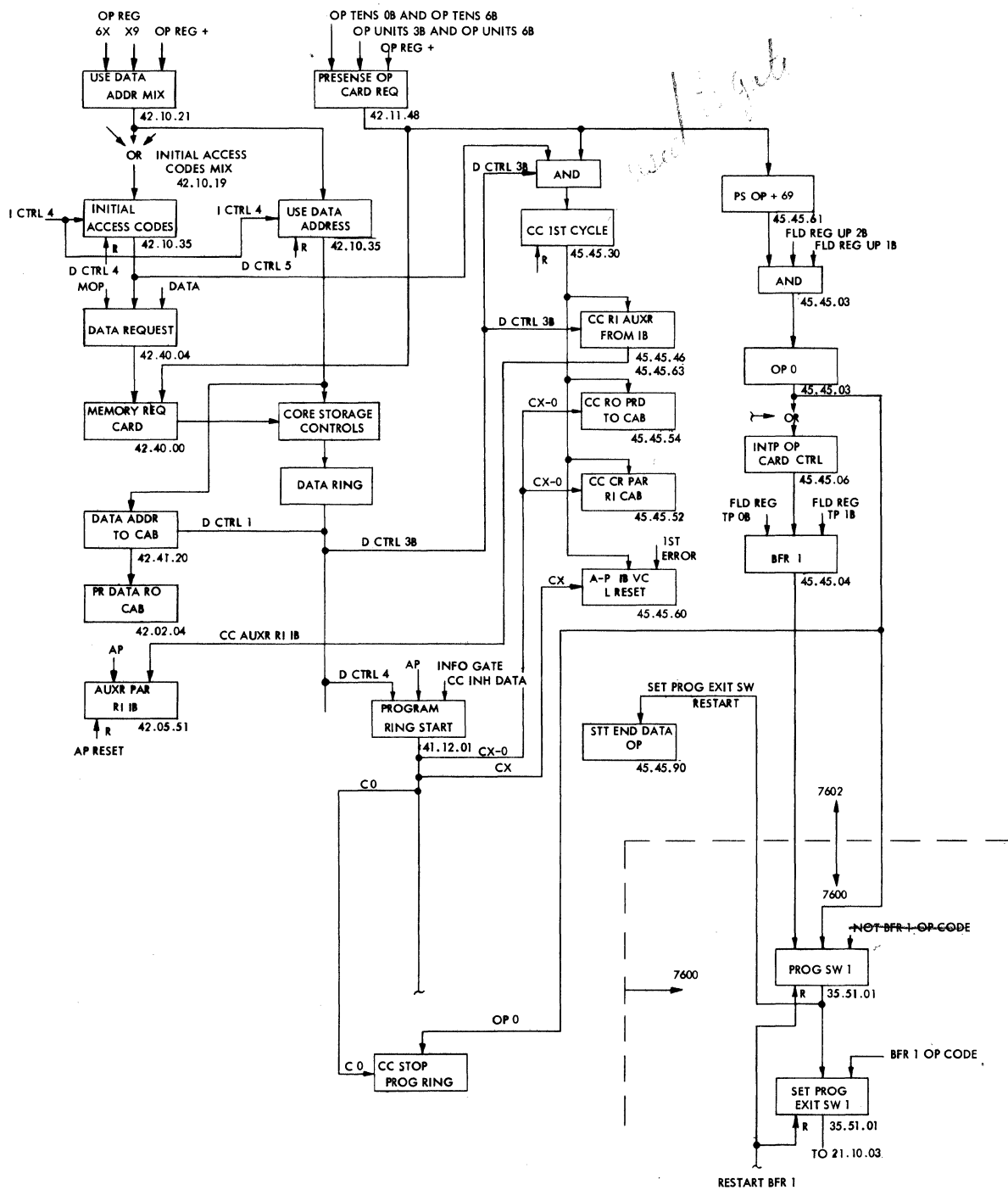


FIGURE 5.12-1. LOGIC FOR SET PROGRAM EXIT SWITCH

ASSUME OP CODE + 0900 11 0200; LOCATION 0200 CONTAINS + 00 0214 0216
LOCATION 0201 CONTAINS - 00 0829 0829

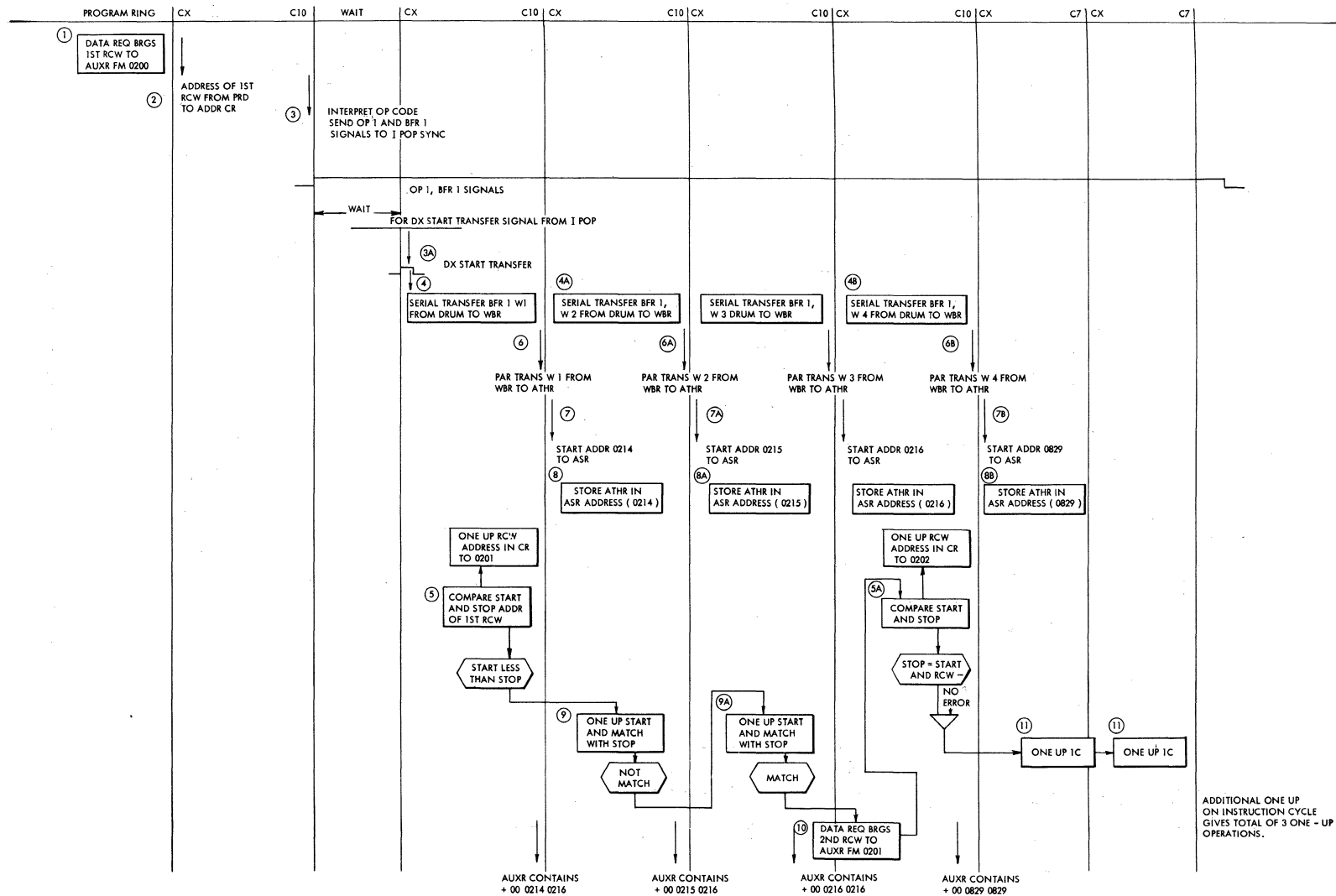


FIGURE 5.12-2. READ OPERATION BLOCK DIAGRAM

⑤ While the first word is being transferred from the drum buffer to WBR, first RCW is tested for the following conditions:

- a. Start address less than stop address. Continue transfer operations.
- b. Start address equal to stop address, and RCW sign plus. Signal data request to secure next RCW and test next RCW.
- c. Start address equal to stop address, and RCW sign minus.

If there has been no error, stop the data transmission and initiate 2 one-up IC operations. Another one-up IC has already taken place during the Instruction Cycle. Thus, the next instruction comes from the address IC + 3.

- d. Start address greater than stop address. This condition should not exist so a RCW Error signal is generated. Only conditions a and c are shown on Figure 5.12-2.

⑥ Word 1 serially transferred from drum to WBR, is parallel-transferred from the WBR to the Athr in preparation for a store in memory operation.

⑦ The start address in the auxiliary register is transferred to the address start register.

⑧ A store-in-memory operation transfers the Athr contents to the start address in the ASR. Note that while the storing of the first word is taking place, the second word is being serially transferred from drum to WBR (See ④A).

⑨ The start address (0214 in the example) is one-upped in the adder and then compared with the stop address by the match circuits (0215 vs 0216 in the example). A Not-match condition results, and the operation continues using the same RCW. The following conditions may result from the match test.

- a. Not Match. Another Match cycle signalled, transfer continues.
- b. Match, RCW Sign Plus. Data Request, ⑩, brings a new RCW to the auxiliary register. Transfer of data continues using the start address of the new RCW to store the next data.
- c. Match, RCW Sign Minus. If no error has taken place, stop the data transmission and initiate 2 one-up IC operations.

④A ⑥A ⑦A
⑧A ⑨A

Repeat same operations as described above. When Start matches Stop, a new RCW is brought to the Auxr by a data-request operation ⑩.

④B ⑥B ⑦B
⑧B

Repeat same operations using the first start address of the 2nd RCW for the store in memory address.

⑪

When the RCW sign is minus, start = stop, and no error exists, 2 One-Up IC operations take place.

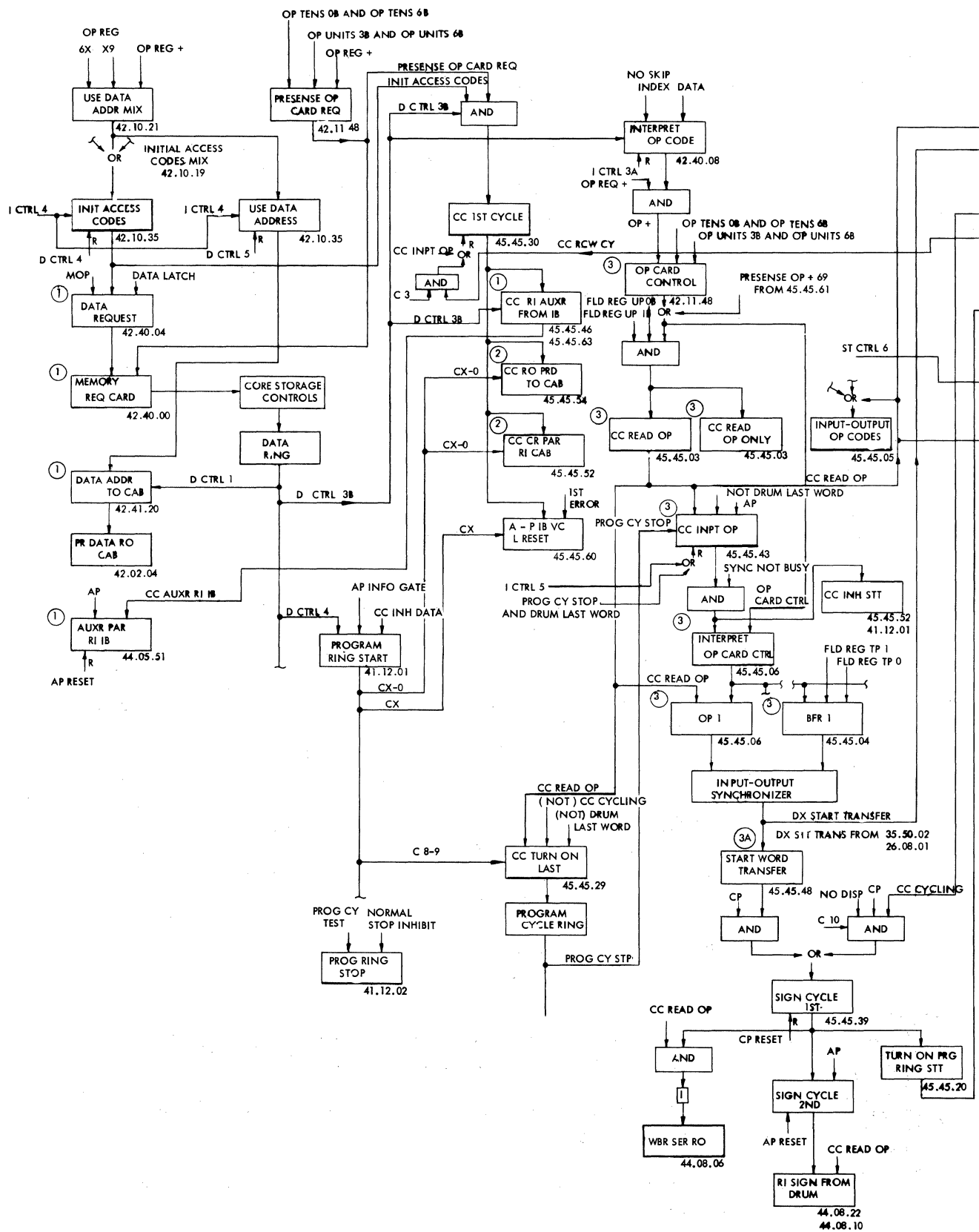


FIGURE 5.12-3A. LOGIC FOR CARD CONTROL READ OP

Circuits for the Card Control Read Operation

No Error Operation. A positive logic circuit summary for the read operation is shown in Figure 5.12-3. The sequence of operation for a typical example is illustrated by Figure 5.12-4. The functional objectives shown on Figure 5.12-2 are keyed to positive logic, sequence chart, and text by circled numbers such as ①

- ① The usual development of initial access codes and use-data-address signals results in a memory request in which the first RCW is brought from the address specified by PRD to the auxiliary register. The D Ctrl 3 B signal switches with Pre-sense Op Card Control and initial access codes to cause a CC 1st Cycle signal. The Data Control 4 signal starts the program ring in the usual manner.
- ② The CC 1st Cycle latch switches with CX-0 to cause the address of the first RCW to be transferred to the Address CR. This is done so that the RCW address may be increased by 1, by the one-up circuits, to create the address of the next RCW to be used.
- ③ The D Ctrl 3B impulse also turns on Interpret Op Code in the usual manner. After switching with Op Reg +, the Op + signal AND's with an Op Register 69 output to turn on Op Card Control. This is further switched with the digit 1 in the 5th position of the instruction (Fld Reg Units Position) to cause both CC Read Op and Input-Output Op Codes signals. Note that all these signals remain on until the Interpret Op Code latch is reset by I Ctrl 3A at the end of the card control operation.

The program cycle ring is turned on by C 8 - 9. The Prog Cycle Stop impulse then turns on CC Interpret Op. This results in the following additional signals which control the read operation.

- a. CC Inhibit Start prevents the start of the program ring by D Ctrl 4 impulses during data request operations until after the end of the read instruction.
 - b. The Op 1 signal is sent to the input-output synchronizer to set up the circuits necessary to transfer the data from buffer to WBR and to signal the start of the next card read cycle.
 - c. The Bfr 1 signal is developed by switching the 1 in the D4 position of the instruction (Fld Reg tens position) with Interpret Op Card Control. This signal is sent to the input-output synchronizer to identify the synchronizer to be used.
- ③A A DX Start Transfer signal at drum Bfr 1, W1, DX time initiates several actions as follows:
- a. The Sign Cycle 1st and Sign Cycle 2nd gates control the read in of the sign to the WBR sign cores.
 - b. Sign Cycle 1st output causes a program ring start. The program ring runs in synchronism with the drum.
 - c. CC Cycling is turned on by DX Start Transfer and remains on until the last data is transferred. The program ring is recycled each C10 time while CC Cycling is on. This results in a repetition of steps ④, ⑥, ⑦, ⑧ shown on Figures 5.12-1 and 2.
 - d. CC RCW Cycle is turned on to cause a comparison of the RCW start and stop addresses.

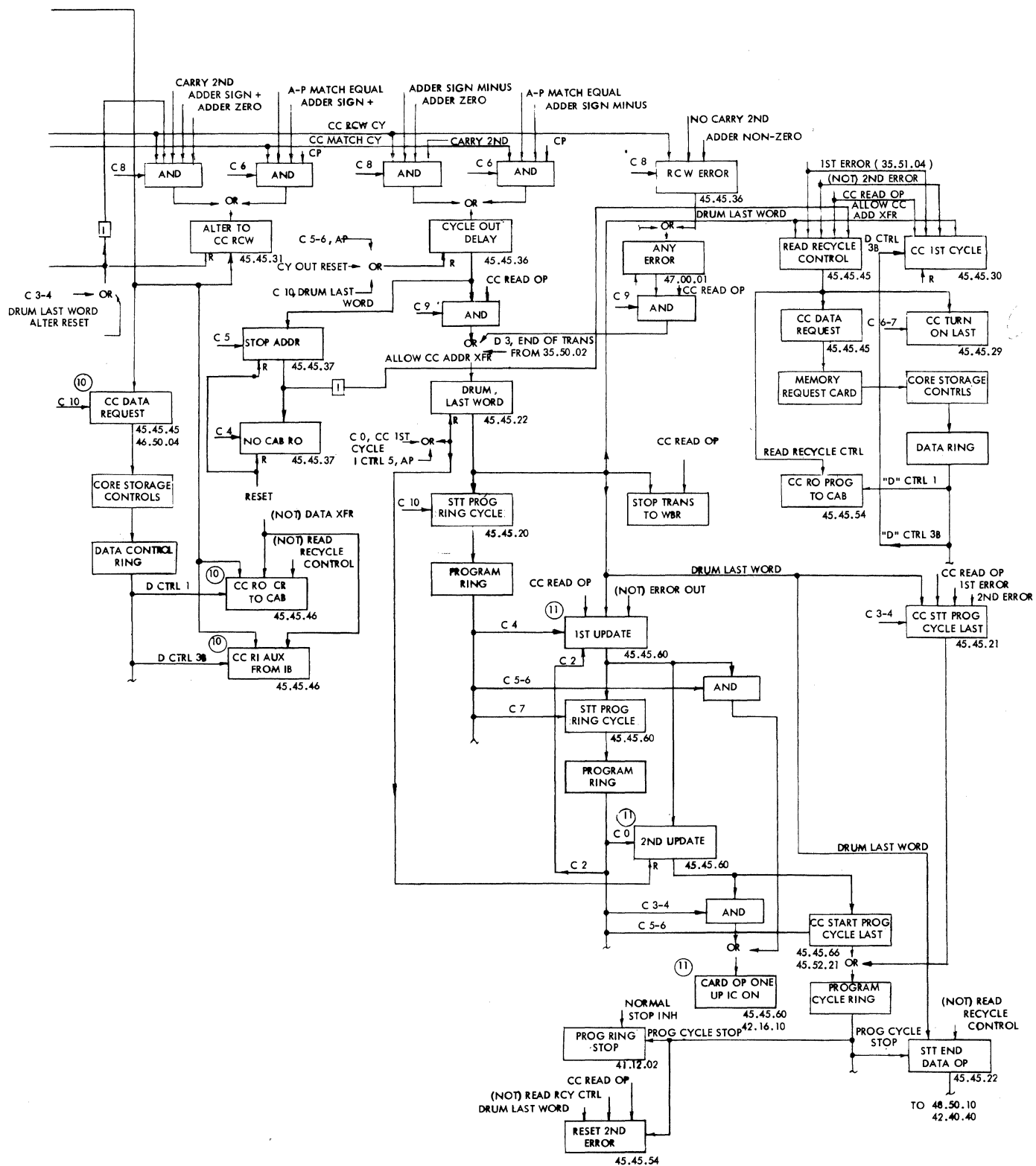
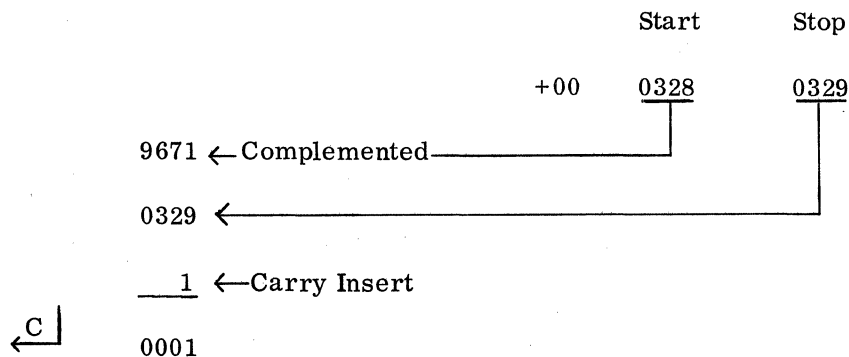
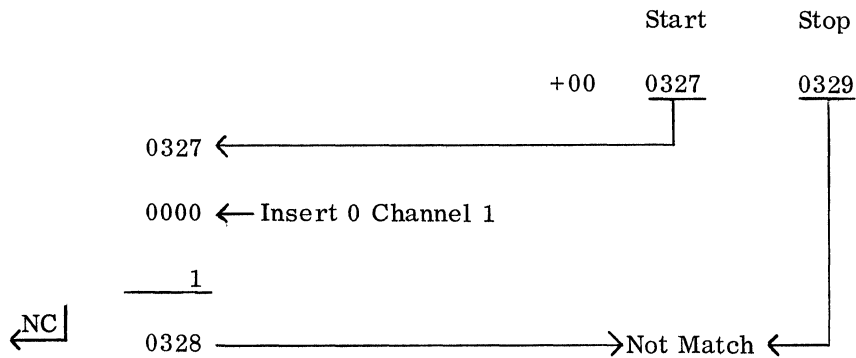


FIGURE 5.12-3C. LOGIC FOR CONTROL READ OP

- ④ The drum serial output is read into the position 9 end of the WBR, and the WBR is shifted left until the sign enters the DS position and the drum units position (drum D0) enters the D9 position of the WBR (Figure 5.12-5).
- ⑥ At the end of the serial drum to WBR transfer, the WBR is parallel-transferred via the arithmetic bus to the Athr.
- ⑦ At the end-of-the-serial transfer of the first word from drum to WBR, the program ring is recycled. The CX-0 output of the recycled program ring causes an address-transfer signal. This results in the transfer of the first start address from Auxr 2 to the address start register.
- ⑧ At C3 time of the recycled program ring, a store-in-memory operation is initiated. The first start address in the ASR addresses core storage via the CAB at St Ctrl 1 time. The arithmetic register is stored in the start address in the usual manner at St Ctrl 3 time.
- ⑨ While the first word read from the drum is being stored and the second word is being transferred to the WBR, the first start address is one-upped and is then compared with the stop address. The Alter to CC Match latch (Figure 5.12-3) controls the operation. This is turned on under two different conditions:
 - a. On a RCW Cycle, when the start address is less than the stop address
The adder output of Carry 2nd and the Adder Non- Zero causes the Alter to CC Match latch to go on at C8 of the RCW cycle.



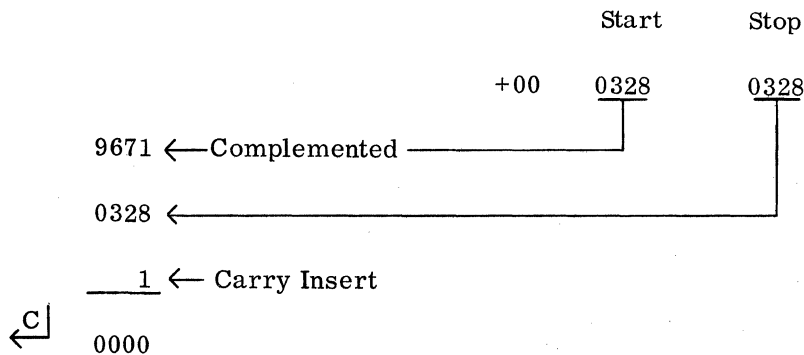
- b. On a match cycle, when the start address is One-Upped, and it fails to equal the stop address, the Alter to CC Match latch is turned on at C6, CP time whenever the A&P Match latch is not on during a CC Match cycle.



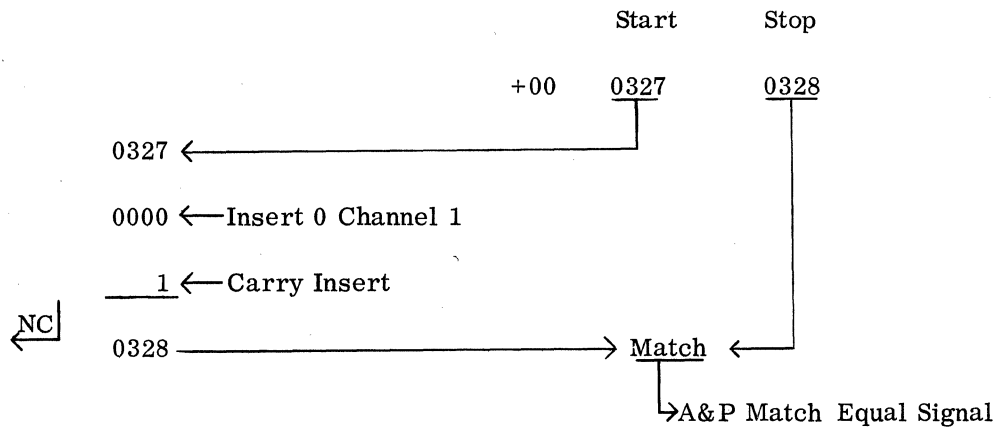
Alter to CC Match turns on the CC Match Cycle latch at CX time. Details of the Match Cycle are discussed in the section on Block Transfer Instructions, Section 5.5.00. Figure 5.5-9 shows the data flow for the one-up start and match start versus stop operations.

- ⑩ When the last start address of a RCW is used, a new RCW is read out from core storage provided that the RCW sign is +. The Alter to RCW latch controls both the data request for the next RCW and the turn on of the CC RCW Cycle latch. Alter to CC RCW is turned on under two different conditions:

- a. On a RCW cycle when Start = Stop, the adder output of Adder Non-Zero and Carry Second cause the turn-on of the Alter to CC Match latch at C8 time of the RCW compare cycle.



- b. On a match cycle when start matches stop after the One-Up operation



alter to CC Match is turned on at C6, CP time when the A-P Match Equal signal occurs on the CC Match Cycle.

2 ways
to turn on
Drum Last wd.

(11)

When the RCW sign is minus, signifying that the last RCW is being used, either of the two conditions in (10) will cause the turn on of the cycle-out-delay latch (Figure 5.12-2). On a read operation, this results in the turn-on of drum last-word at C9 time. Note that drum-last-word may also be turned on by a D3, end-of-transfer signal from the IPOP Synchronizer. This occurs at W 16, D3 time, when the end of the drum buffer comes before the last RCW is sensed.

The program ring is recycled at C10 time of drum-last-word. At C4 time of the recycle, 1st Update is turned on, provided no error has been sensed. While 1st Update is on, a C5-6 signal begins at IC one-up operation.

The program ring is again recycled at C7 time of the 1st Update operation. The 2nd Update latch comes on at C0 time to control the C3-4 time start of the second IC one-up operation. The program cycle last ring is started at C5-6 time of the second update cycle. The program cycle stop output of the ring turns on end-data operation to cause an instruction cycle to start. The IC was updated once during the instruction cycle so that the next instruction is taken from I + 3 when the data is read without error.

Error Recycle Operation. When an error occurs during the transmission of data from the reader to the WBR, a 1st Error signal from logic 35.51.04 switches with drum-last-word, Read Op and (not) 2nd Error to turn on the read recycle control (Figure 5.12-3C). This results in a second transfer of data from drum to WBR under RCW control. Drum-last-word switches with C10 to cause a program ring recycle in the usual manner. Updating does not take place because 1st Update does not come on due to the error. The read recycle control output causes a memory request operation in which the first RCW, in the address specified by the address in PRD, is read out to the auxiliary register. The D Ctrl 3B signal switches with Read Op, 1st Error, (Not) 2nd Error and Allow CC Add, Xfr to turn CC 1st Cycle back on. The operation continues as previously described.



Figure 5.12 - 4

214A

At the end of the read recycle operation, updating occurs in the usual manner if a second error has not taken place. If the error remains uncorrected at the end of the read recycle, the program cycle last ring is started, without updating, by switching C3-4, Drum Last Word, 1st and 2nd Error and Read Op. This results in an end-data-op signal. The IC is updated just once, during the instruction cycle, and the next instruction comes from IC+1. *done*

End of File Operation. When information from the last card is transferred from input buffer to core storage, an end-of-file latch is turned on (See logic 45.45.80). With EOF on, the next read command is interpreted as a No Op, the EOF latch is reset, the IC is advanced by two and the program continues. The additional logic required for EOF is shown on logic 45.45.80, and the sequence of operation is illustrated by Figure 5.12-6.

5.12.03 Write Operation

Introduction

A two in the Digit 5 position of the Card Control Op code causes a write or punch operation. A block transmission of up to 16 words, as defined by record control words, takes place from core storage to the output buffer specified by the digit in position 4 of the instruction. If the record control words define fewer than 16 words, the remaining buffer words are filled in with zeros and assigned positive signs. As with the read operation, if the data transmission is valid, the IC is advanced by 3 and a punch or print cycle is performed. If the data transmitted is invalid, the IC is advanced by 1 and no punch or print operation takes place.

The functions performed during each cycle of the program ring for a specific example of a write operation are shown in Figure 5.12-7. After the pre-sensing of the card-control operation, a data request takes place as with any other initial access code. The RCW in the core storage location indicated by the address in PRD is brought to the auxiliary register ①.

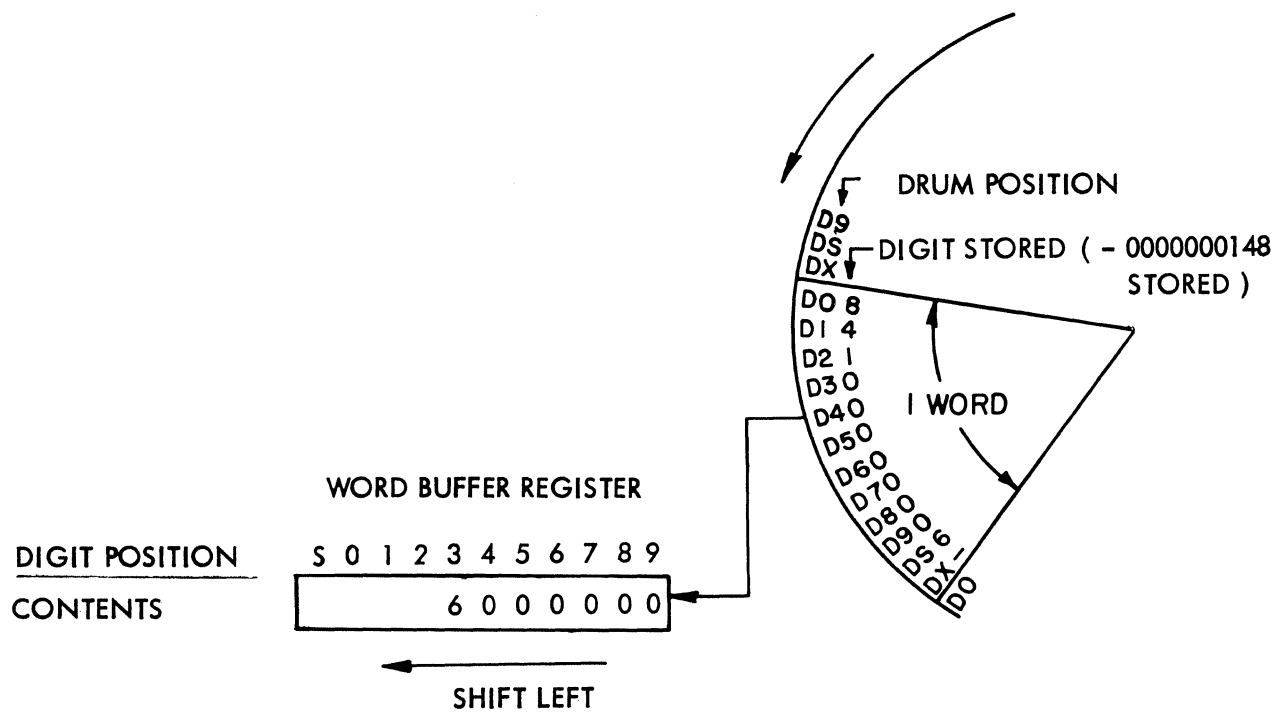
- ② The address of the first RCW is parallel-transferred from the PR D to the Address Control Register (CR). After each use of a new RCW, the address in the CR must be increased by 1 by the one-upper circuits in preparation for reading out the next RCW. This is shown in step ④ on Figure 5.12-7.
- ③ Before the first word is transferred from core storage to WBR in preparation for printing or punching, the RCW start address must be compared with the stop address. The results of the comparison may be any of the following:
 - a. Start address less than stop address. Continue operation with a data request to transfer an output word from core storage to the arithmetic register.
 - b. Start = Stop and RCW sign plus. Continue operation with a data request to transfer the next RCW from core storage to the auxiliary register.
 - c. Start = Stop and RCW sign minus. Transfer one word from core storage to Athr to WBR to IPOP Synchronizer and end the operation with two updating operations.
 - d. Start greater than stop. This condition should not occur so a RCW Error signal occurs.

- ④ While the start and stop addresses of the RCW are being compared, the address of the last RCW used is being upped by 1 by the CR one-upper circuit.
 - ⑤ The first start address to be used is parallel-transferred from Auxr 2 to the ASR.
 - ⑦ A data-request operation reads out the data in the location designated by the start address, in the ASR, to the arithmetic register.
 - ⑧ After the end of the data request, the Athr contents are parallel-transferred to the WBR. This is in preparation for the serial transfer to the drum which follows after the Op code development and the receipt of the DX start transfer signal from the IPOP synchronizer.
 - ⑥ While the data request for the first output word is being accomplished, the adder is used to one-up the first start address in Auxr 2. At the same time, the match circuit is used to determine whether the one-upped start address matches the stop address. There are three possible results of the test of the match circuit output, as follows:
 - a. Not Match. Another one-up and match operation occurs in this case after the delay occasioned by waiting for the drum to turn to the Buffer 1, W1, DX position.
 - b. Match and RCW sign plus. A data request for another RCW takes place.
 - c. Match and RCW sign minus. End of both data request operations and serial transfers to drum are signalled; one-up IC and end-data operation occurs.
 - ⑨ At the time the operation signals are sent to the IPOP synchronizer, the following have already occurred:
 - a. The first word to be transferred to the drum is in the WBR.
 - b. The start address has been one-upped to form the second start address.
 - c. A match operation has resulted in the turn-on of a latch to indicate whether a new RCW should be requested, or whether the start address should be one-upped again.
 - ⑩ When the DX Start Transfer signal is returned to the A-P circuits at Bfr 1, W1, DX time, the program ring is recycled and the serial transfer of data from WBR to drum begins.
-
- 5A

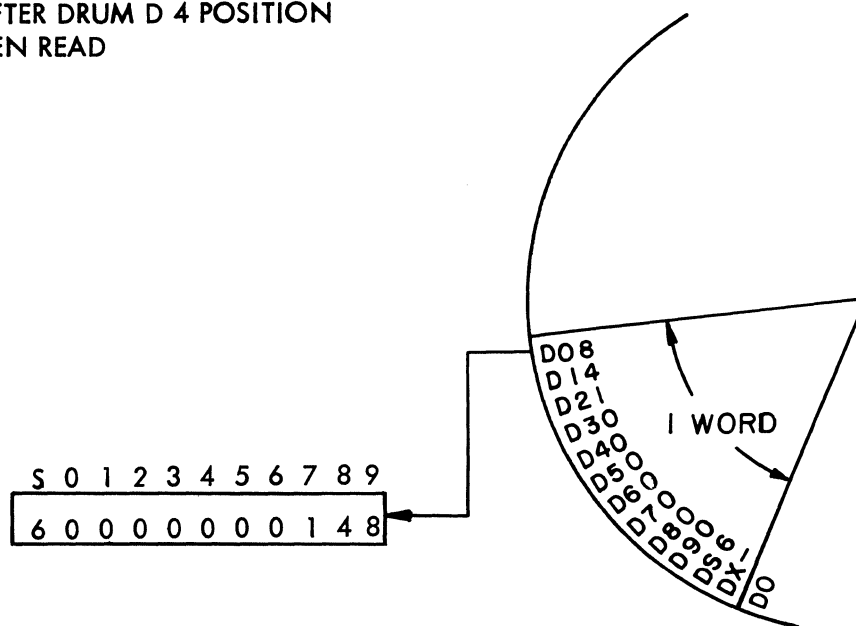
7A

8A

While the serial transfer from WBR to drum is going on, a parallel-transfer places the start address in the ASR and a data request brings the second output word to the Athr from the location specified by the ASR. After the end of the serial transfer from the WBR, a parallel-transfer places the second output word in the WBR.



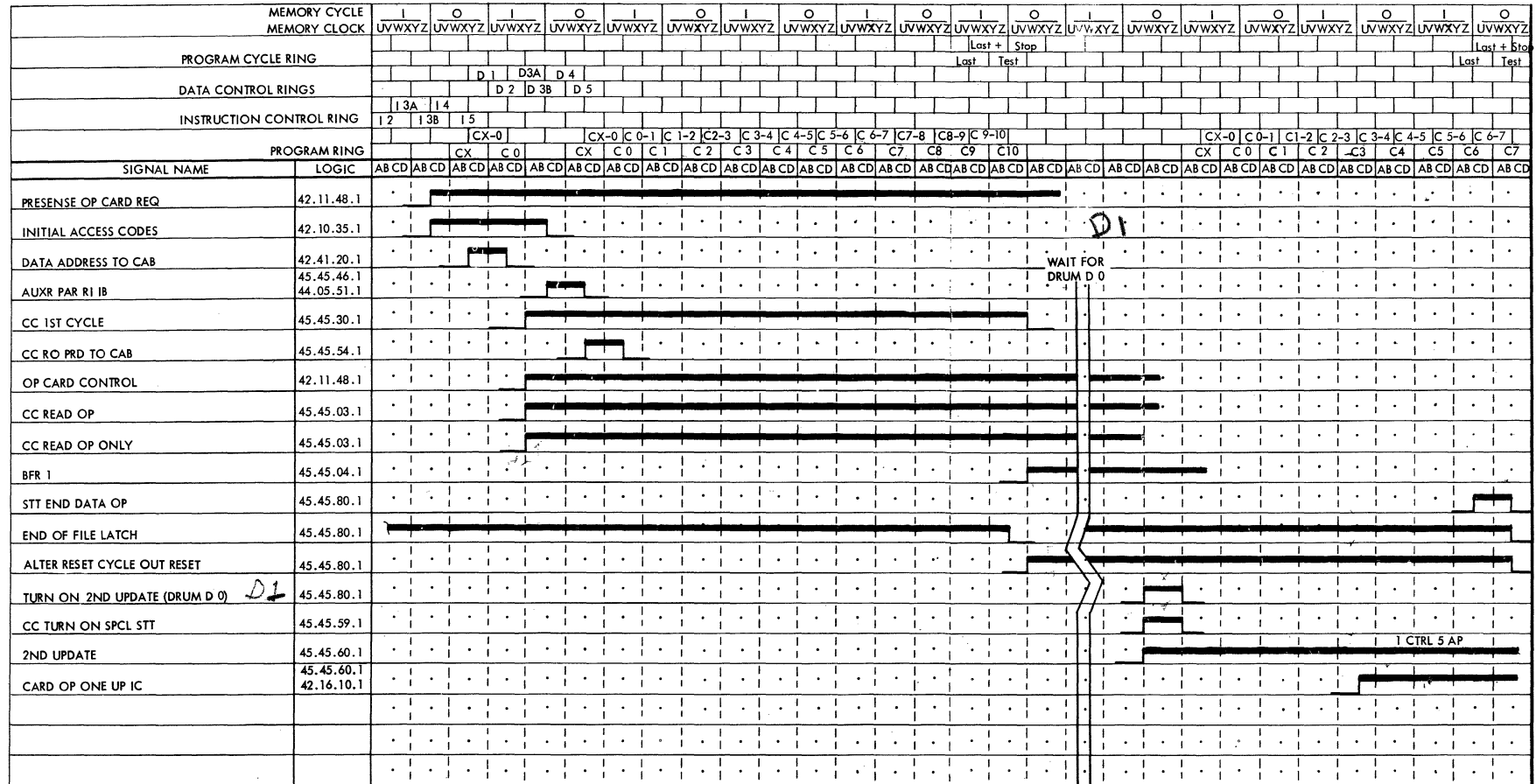
(a) WBR AFTER DRUM D 4 POSITION HAS BEEN READ



(b) WBR FILLED

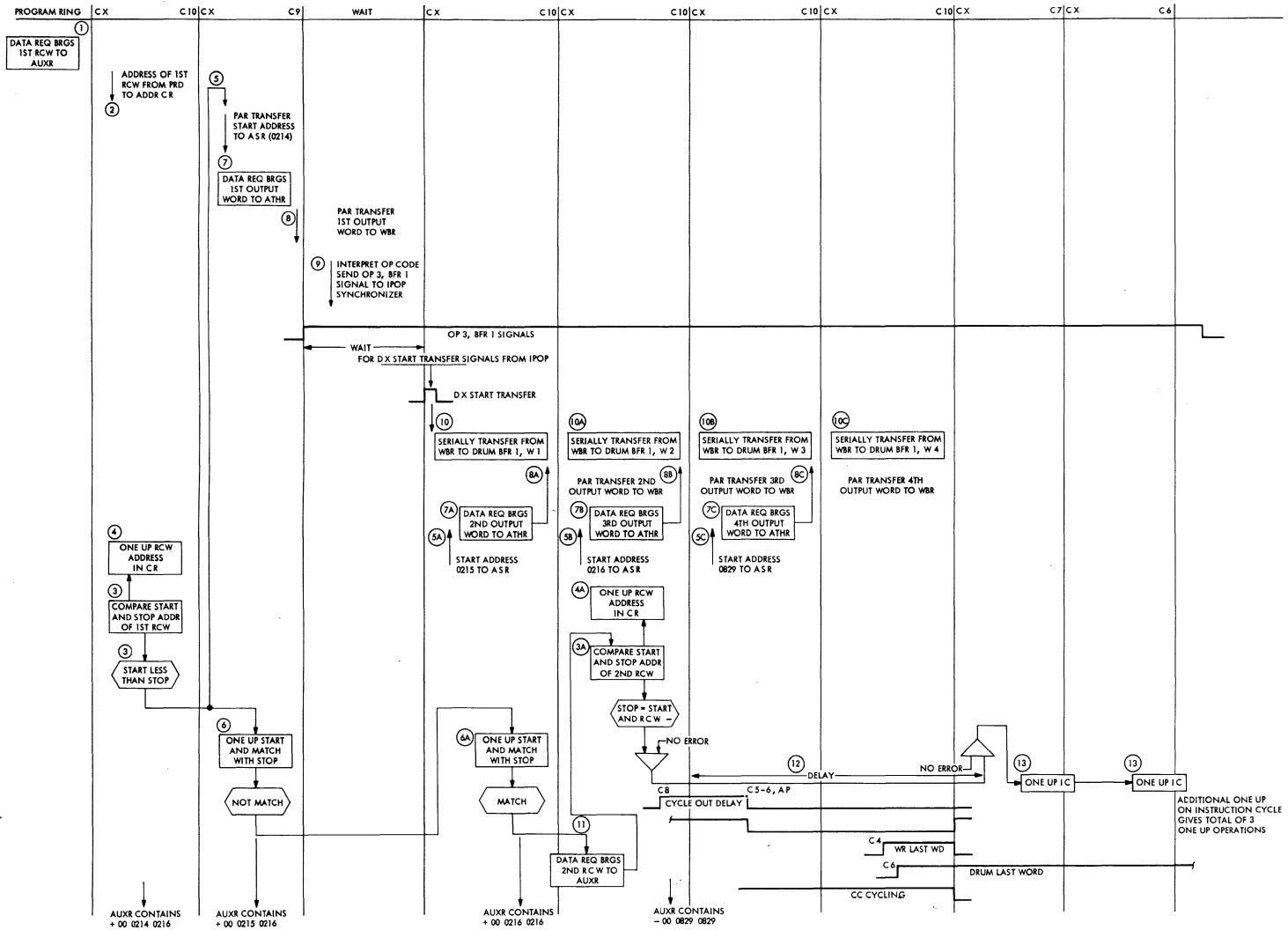
FIGURE 5.12-5. DRUM TO WBR TRANSFER

FIGURE 5.12-6. END-OF-FILE SEQUENCE



ASSUME OP CODE = 69 00 12 0200 LOCATION 0200 CONTAINS + 00 0214 0216
LOCATION 0201 CONTAINS - 00 0829 0829

FIGURE 5.12-7. WRITE OPERATION BLOCK DIAGRAM



⑥⑪ The one-up and match operation results in a match and the ensuing data request to bring the second RCW to the Auxiliary Register.

⑦⑩ Repetition of ⑦ and ⑩

③ This time the comparison of the RCW start and stop address results in an equal indication. Because the sign of the RCW is minus, and no error has occurred, the end of the operation may be signalled. A delay, ⑫, is necessary because the last word designated by the RCW must be data requested for Core Storage and serially transferred to the drum before the program ring can be recycled for the one-up IC operations, ⑬.

Circuits for the Card Control Write Operation (Figures 5.12-8, 9)

① The analysis of Op code + 69 results in the development of initial access codes and use-data-address signals. This results in the usual memory request in which the first RCW is brought from the address specified by PRD to the auxiliary register. The D Control 3B signal switches with pre-sense op card control and initial access codes to cause a CC 1st Cycle signal. The D Control 4 signal starts the program ring in the usual manner.

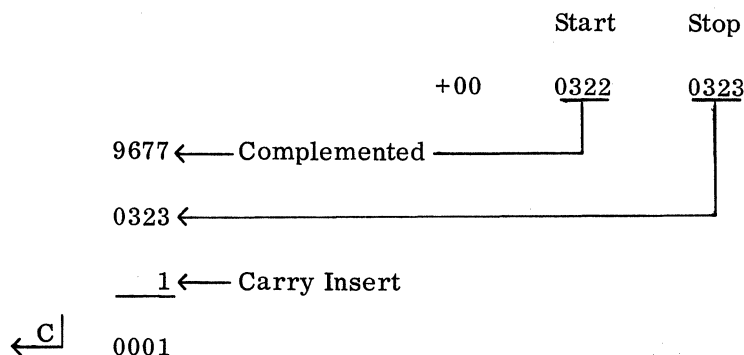
② The CC 1st Cycle latch switches with CX-0 to cause the address of the first RCW to be transferred to the Address CR. This is done so that the RCW address may be increased by 1 by the one-up circuits to create the address of the next RCW to be used.

③ The D control 3B, Initial Access Codes, Pre-sense op card request signal also turns on the CC RCW Cycle latch. This latch controls the data flow switching required to compare the RCW start address with the stop address. The following functions are performed by the switching:

- a. Advance Auxr 2 and Auxr 3 to right.
- b. Read-out Auxr 2 to Ch 2 and Auxr 3 to Ch 1.
- c. Regenerate Auxr 2 and Auxr 3.
- d. Complement Add Auxr 2 to Auxr 3; carry insert.

As a result of this operation, Auxr 2 (start) and Auxr 3 (stop) remain unchanged, but the adder output furnishes signals which are tested to determine the relative size of the start and stop addresses as follows:

- a. Start less than Stop



At C8 time of the CC RCW Cycle, Adder-Non-Zero and Carry 2nd are switched together to turn on Alter to CC Match. This in turn results in a CC Match Cycle, (6). The match cycle one-ups the start address and compares it with the stop address. Alter to CC Match also turns on Write Cycle in 1st at C10 time of the 1st program ring run on a write cycle. The CC Cycling latch (Figure 5.12-8C) is not turned on until after the op code is analyzed and the DX start transfer signal is returned to the A&P unit. Thus, the No CC Cycle line is up and Write Cycle in 1st can be turned on. The Write Cycle in 1st latch controls functions (5), (7), (8), (9). The Write Cycle in 1st latch remains on until turned off by CC Cycle. Through Write Cycle in 1st may remain on for some time, functions (5), (7), (8), and (9) are performed only once because the program ring makes only one run through its cycle before the wait for DX Start Transfer begins. The program ring is recycled by switching C10 with Write Cycle in 1st. The program cycle last ring is started at C7-8 time. Program cycle test then stops the program ring.

- b. Start equal to Stop and RCW Sign plus. In this case RCW Sign plus, Adder Zero, Carry 2nd, C8 and RCW Cycle switch to turn on Alter to CC RCW (Figure 5.12-8B). This results in a data-request operation to read-out the next RCW to the auxiliary register. In addition, Write Cycle in 1st is turned on provided that CC Cycling is not on.
- c. Start equal to stop and RCW sign minus. When the conditions in (b) exist but the RCW sign is minus, cycle-out delay (12), begins and the IC is one-upped twice (13), provided no error exists.
- d. Start greater than stop. This condition should never occur so a RCW Error is signalled by switching C8, No-Carry 2nd, Adder Non-Zero and CC RCW Cycle, Figure 5.12-8B. Note that the sign of the RCW is transferred to the address sign latches at C3-4 and C4 time of the RCW Cycle.

- (4) While the comparison of start and stop addresses is taking place during the RCW cycle, the address of the current RCW is one-upped in the CR to form the address of the next RCW to be used. This is a typical one-up operation similar to the one-up IC which takes place during each instruction cycle.

- (5), (7), (8) The Write Cycle in 1st signal switches with CX-0 (Figure 5.12-8C) to cause an address transfer signal. This results in the transfer of the first start address from Auxr 2 to the ASR (5). Next, a Data Xfr operation is started at C3 time to request from memory the data word indicated by the start address in the ASR (7). After the data from core storage enters the arithmetic register, it is parallel-transferred to the WBR in preparation for the serial transfer to the drum buffer.

Note that CC Inhibit Stt (Figure 5.12-8C) is turned on by Write Cycle in 1st. This prevents the program ring from starting at D Control 4 time during the data request.

- ⑨ Interpret Op Code and Op + signals are developed in the usual way as shown in Figure 5.12-8A. Op + switches with Op Register 69 as shown on 5.12-8C to cause an Op Card Control Signal. This switches further with the 2 in the Digit 5 position of the instruction to form the following signals:

- a. CC Write Op
- b. CC Write IPOP
- c. All CC Write Ops
- d. Input-Output Op Codes

After the first word to be transferred to the drum for the write operation has entered the WBR, a program-cycle-stop impulse turns on CC Interpret Op. This in turn results in the development of an Op 3 signal if CC Write Op is on and a Bfr 1 signal if the 4th position of the instruction contains the digit value 1. Both Op 3 and Bfr 1 signals are sent to the IPOP synchronizer. A DX Start Transfer signal is returned when the drum has turned to the buffer 1 position and is ready to receive data from the WBR.

- ⑩ The DX Start Transfer signal switches with a CP (Figure 5.12-8C) to turn on Sign Cycle 1st. This causes a WBR Read-Out Sign gate and starts the program ring.

The DX Start Transfer signal also turns on CC Cycling (Figure 5.12-9). CC Cycling remains on until the end of data transfer and control operations ⑤A, ⑦A, ⑧A, and ⑩A. The program ring is recycled each C10 time by CC Cycling until CC Cycling is turned off by CX, Drum Last Word.

- ⑥ On each program ring cycle of each RCW, except the first, a one-up and match operation takes place. Details of the match cycle are discussed in the section on Block Transfer, Section 5.5.00. Figure 5.5-9 shows the data flow for the one-up start and match start versus stop operations. The match cycle may result in any of the following:

- a. Not A&P Match latch. The start address has not yet been increased so that it equals the stop address so another CC Match cycle must be taken. Figure 5.12-8B shows how CC Match Cycle switches with C6, CP and (Not) A&P Match latch to turn on Alter to CC Match. The CC Match Cycle latch comes on again at CX time and another one-up and match occurs.
- b. A&P Match Equal latch on and RCW Sign +. The start address matches the stop address so another RCW must be read out of core storage. A&P Match Equal Addr Sign +, CC Match and D6, CP switches as shown on Figure 5.12-8B to turn on Alter to CC RCW. A data request operation brings the next RCW to the auxiliary register and the CC RCW Cycle latch is turned on to cause the comparison of start and stop addresses as described in ③ previously.
- c. A&P Match Equal latch on and RCW Sign -. The start address equals the stop address and the last RCW has been used. Figure 5.12-8B shows how CC Match Cycle switches with C6, CP, Addr Sign Minus and A&P Match Equal to turn on Cycle Out Delay ⑫.

- ⑫ The circuits for the cycle-out delay sequence are shown on Figure 5.12-8B, and a sequence chart is shown on Figure 5.12-7. As described previously, the cycle-out-delay latch is turned on at C8 time of the RCW Compare Cycle. At C5 time of the next cycle, the stop-address latch comes on. This signal is inverted to form the Xfr Write Data signal. Thus the Xfr Write Data signal is down while the stop address latch is on. The Xfr Write Data signal is down during the last serial transfer operation. This prevents a Data Xfr Operation (Figure 5.12-8C) 7A.

During the last serial transfer operation the No CAB RO and Writing Last Word signals come on at C4 time. At C6 time, Drum Last Word is turned on. This is inverted (Figure 5.12-8C) 8A to prevent Athr to WBR transfer.

The CC Cycling latch is not turned off until CX, Drum Last Word time. This allows the controls for serial transfer from WBR to drum to operate during the last full cycle of the program ring.

- ⑬ As shown in Figure 5.12-8B, the Drum Last Word latch comes on at C6 time when the last RCW is used; or at D3 drum time when the last word of the buffer is filled. In either case, provided an error has not occurred, two one-up IC operations take place. The updating works in the same manner as on a read operation.

Drum Last Word switches with CX to signal start end-of-data to the input-output synchronizer. This results in the writing of plus zeros in the remaining words of the buffer.

If an error has occurred, a C0 impulse starts the Program Cycle Last Ring, and program cycle stop causes an end data op signal. If no error has occurred, the Program Cycle Last Ring is started at C5-6 of the second update cycle (Figure 5.12-8B).

5.12.04 Write or Punch Invalid Operation

Introduction

By means of this instruction, records containing known non-valid characters may be printed or punched. The core storage to synchronizer validity check signal is suppressed for the designated buffer during the transmission of data. An early timed error write impulse (EWI) is available on the output machine control panel for the control of selectors, etc. The instruction counter is advanced by one after the data is transmitted to the drum buffer.

Circuits

Circuit operation is substantially the same as on the write operation. Figures 5.12-7, 8, 9 may be used for the circuit operation and sequence, keeping in mind that the following signals are developed for the error-write operation.



FIGURE 5.12-8A. LOGIC FOR CARD CONTROL WRITE OP

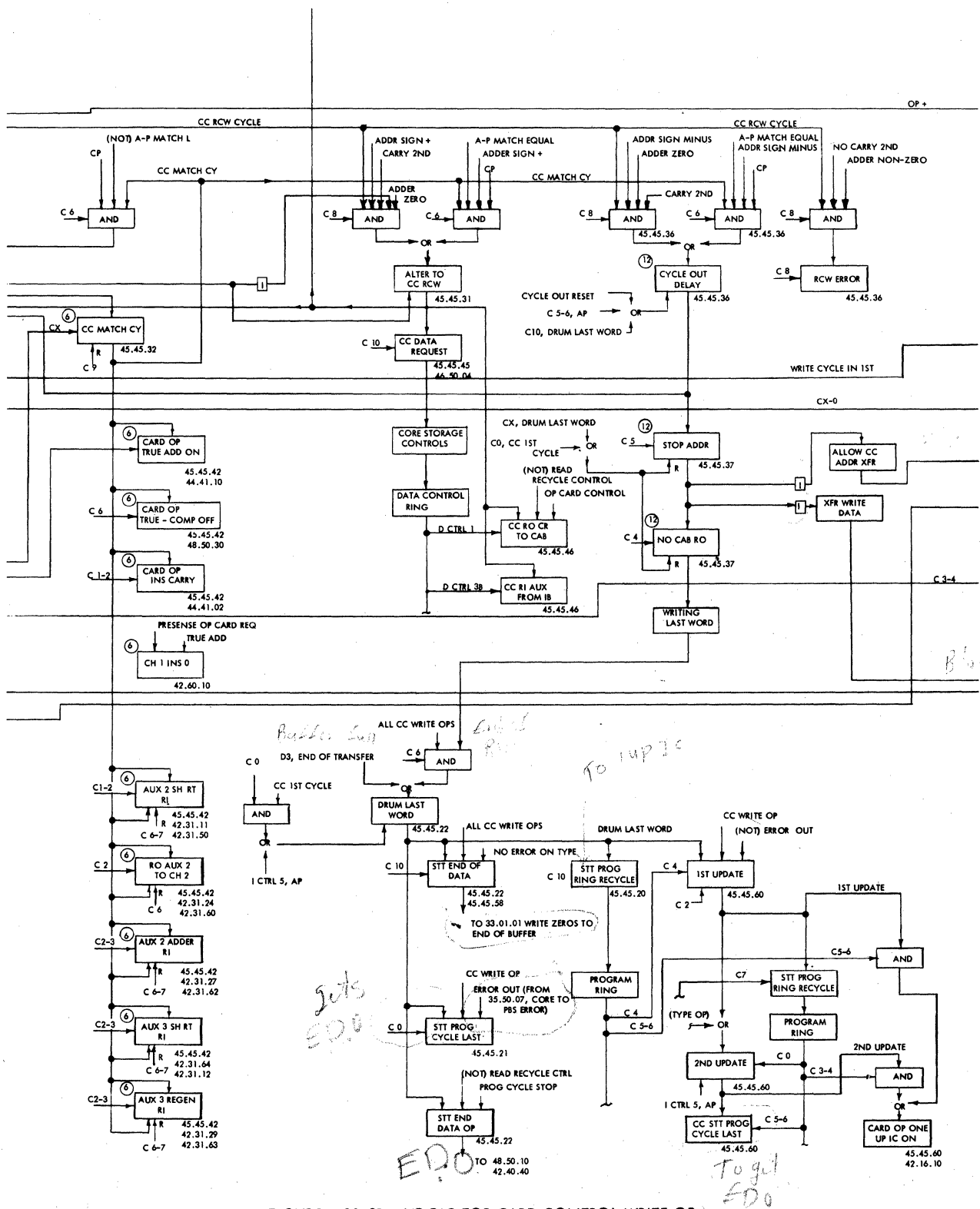


FIGURE 5.12-8B. LOGIC FOR CARD CONTROL WRITE OP

<u>Signal</u>	<u>Logic</u>
CC Error-Write Op	45.45.03
Op 2 or Op 3	45.45.05
Input-Output Op Codes	45.45.05
CC Write IPOP	45.45.05
All CC Write Ops	45.45.05
Op 2	45.45.06
Buf 1, Buf 2 or Buf 3	45.45.04

Note that only the CC Write Op signal is not developed for the error-write operation. This prevents the turn on of 1st Update (Figure 5.12-8B) in case the data transmitted happens to be valid.

The next instruction is always taken from the address IC + 1 so the End Data operation takes place without updating. Drum Last Word is switched with CX-0 and CC Error Write Op on logic 45.45.21 to cause a start-program-cycle-last signal. The program-cycle-stop impulse switches with Drum Last Word (Figure 5.12-8B) to cause an end-data-op signal. The program-cycle-test signal stops the program ring in the usual way (Logic 41.12.02).

EWI of error punchout

Control of the WBR to PBS validity check signal is accomplished in the input-output synchronizer. The WBR to PBS validity check signal is sent to the output machine for conversion to an early timed control panel impulse to allow format control (or off-set stacking if punching) of the invalid data. (The A&P IB VC latch is reset on logic 45.45.60 by switching Drum Last Word and CC Error Write Op). The Control of the WBR to PBS Validity Check signal is discussed further in Section 3.7.00 of the Input/Output Manual of Instruction (Volume 1).

5.12.05 Type Operation (Figures 5.12-10, 11)

Introduction

This operation causes the typing out of any part or all of core storage under control of record control words. All computing by the 7070 is suspended until the contents (of the location defined by the start address of the last RCW) is typed. An automatic carriage return is performed when the end of each writing line of the typewriter is reached.

Record Control Word Error Operation. If the first RCW used in a program controlled typing operation has any of the following listed errors, the system stops immediately and the contents of the Instruction Counter (IC + 1) and the program register type out.

- a. Start greater than stop.
- b. Alpha sign
- c. Illegal address
- d. Invalid digit

If any of these conditions are detected on subsequent record control words, the following sequence of operations takes place:

1. Finish typing block of words defined by the previous RCW.
2. Type red* after the last digit or character typed.
3. Program stop
4. Carriage return
5. Type out contents of instruction counter, IC + 1, and program register (Type instruction)
6. Carriage return

Control Register A Error Operation. Detection of either an illegal address or an invalid digit in the control register causes the 7070 to operate in the same manner as as described under RCW operation.

Invalid Information Error Operation.

1. When an invalid digit is detected between core storage and the WBR, "bit printing" in red of the invalid digit takes place. Typing continues until the end of the block defined by the last RCW. The next instruction is taken from the address in the IC.
2. When an invalid digit is detected between the WBR and the typewriter, "bit printing" in red of the invalid digit takes place. At the completion of the message, the next instruction is taken from IC + 1. The type operation words in a manner similar to the write operation. Figure 5.12-10 illustrates the principle involved. After the transfer of the first word to be typed to the WBR and the development of the operation code signals the console, the program ring is stopped and the machine waits for a Con Stt Word Transfer signal at the beginning of the S5 portion of the output buffer. The program ring is restarted, and the transfer of data begins. The ring is recycled each D10 time and the transfer of data continues until the ten words of the type buffer are filled and a S5, W10. D1 time impulse turns the ring off and stops the data transmission.

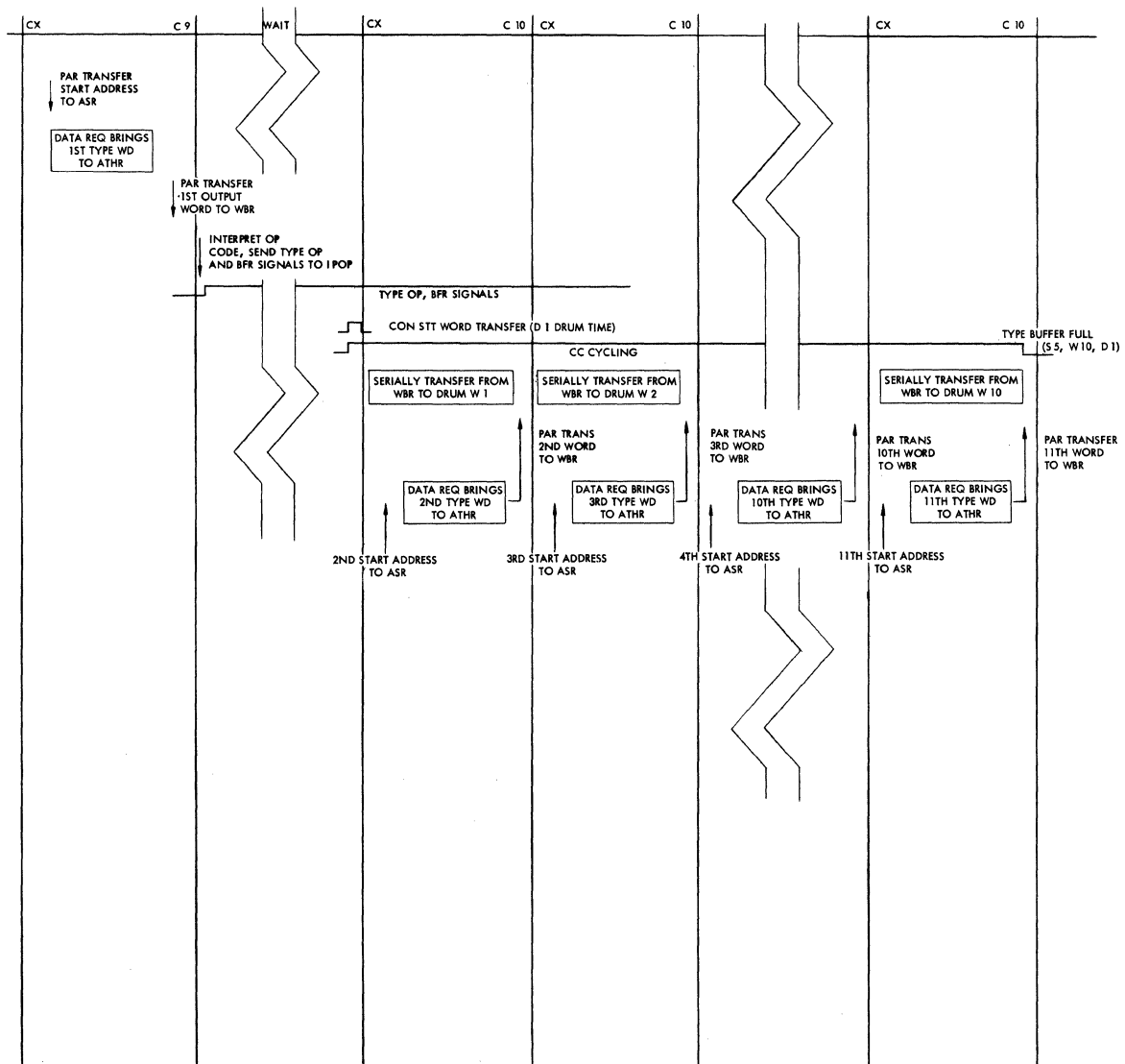


FIGURE 5.12-10A. SEQUENCE FOR TYPE

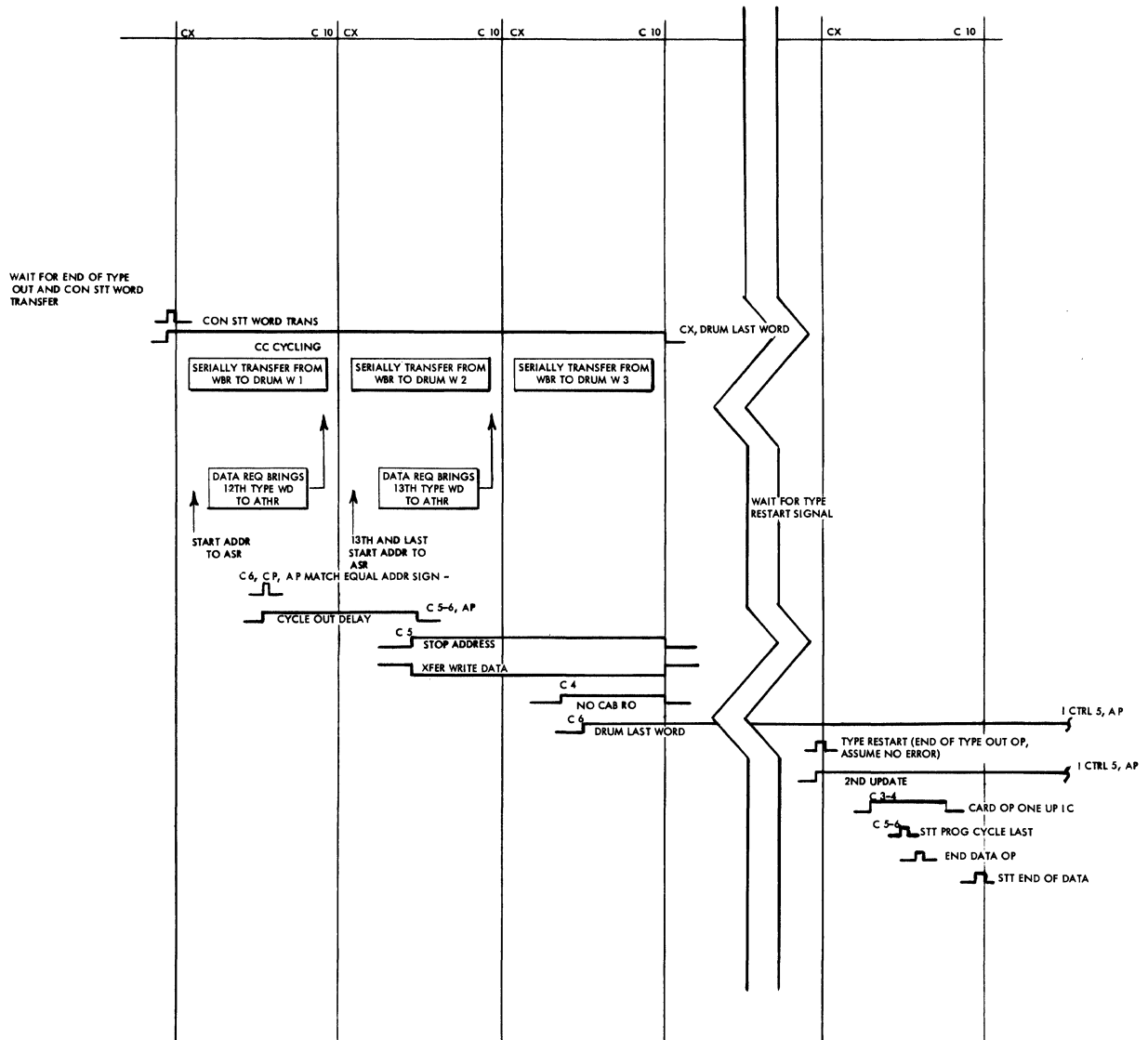


FIGURE 5.12-10B. SEQUENCE FOR TYPE

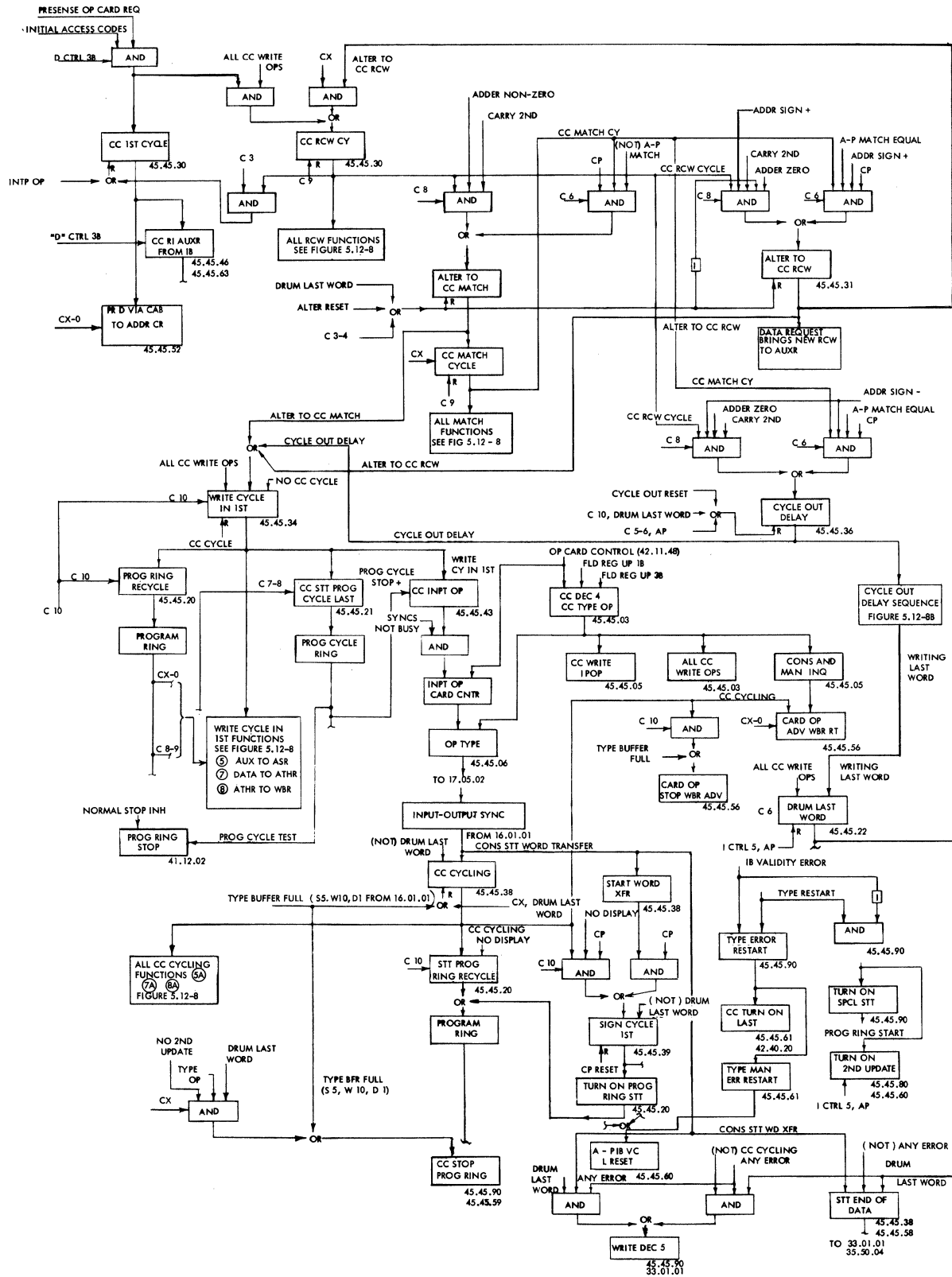


FIGURE 5.12-11. LOGIC FOR CARD CONTROL TYPE OP

The ten words of the type buffer are now typed out and a Con Stt Word Transfer signal is returned to the A&P unit when the buffer is emptied and the drum has turned to the beginning of S5. The next group of ten words is now transferred. This procedure continues under control of the RCW system until the start and stop addresses of the last RCW (minus sign) are equal. The usual cycle out-delay sequence is signalled. The end-data-operation signal does not occur until after the last word has been typed.

Type Operation Circuits

No Error Operation. Figure 5.12-11 shows the basic control circuits for the type operation. Much of this circuitry operates in the same manner as on the write operation. The main differences, in the no-error operation, are in the control of the CC Cycling latch and the program ring. The circuits, as illustrated in Figure 5.12.11, follow directly from the functional principles discussed in the preceding introduction.

Error Operation. Figures 5.12-12, 13 show the sequence of operation for errors in the first and succeeding record control words.

MEMORY CYCLE		MEMORY CLOCK														
		I	O	I	O	I	O	I	O	I	O	I	O			
		UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ	UVWXYZ			
DATA CONTROL RING		D1	D 3A	D 4												
		D 2	D 3B	D 5												
PROGRAM RING		CX-0	C0-1	C1-2	C2-3	C3-4	C4-5	C5-6	C6-7	C7-8	C8-9	C9-10				
		C 0		CX	C 0	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	
SIGNAL NAME	LOGIC	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD
CC 1ST CYCLE	45.45.30.1															
CC RCW CYCLE	45.45.30.1															
COMPARE START AND STOP ADDRESSES OF 1ST RCW																
RCW ERROR	45.45.36.1															
ANY ERROR	47.00.01															
RCW ERROR ON WRITE	45.45.90.1															
STOP ADDRESS	45.45.37.1															

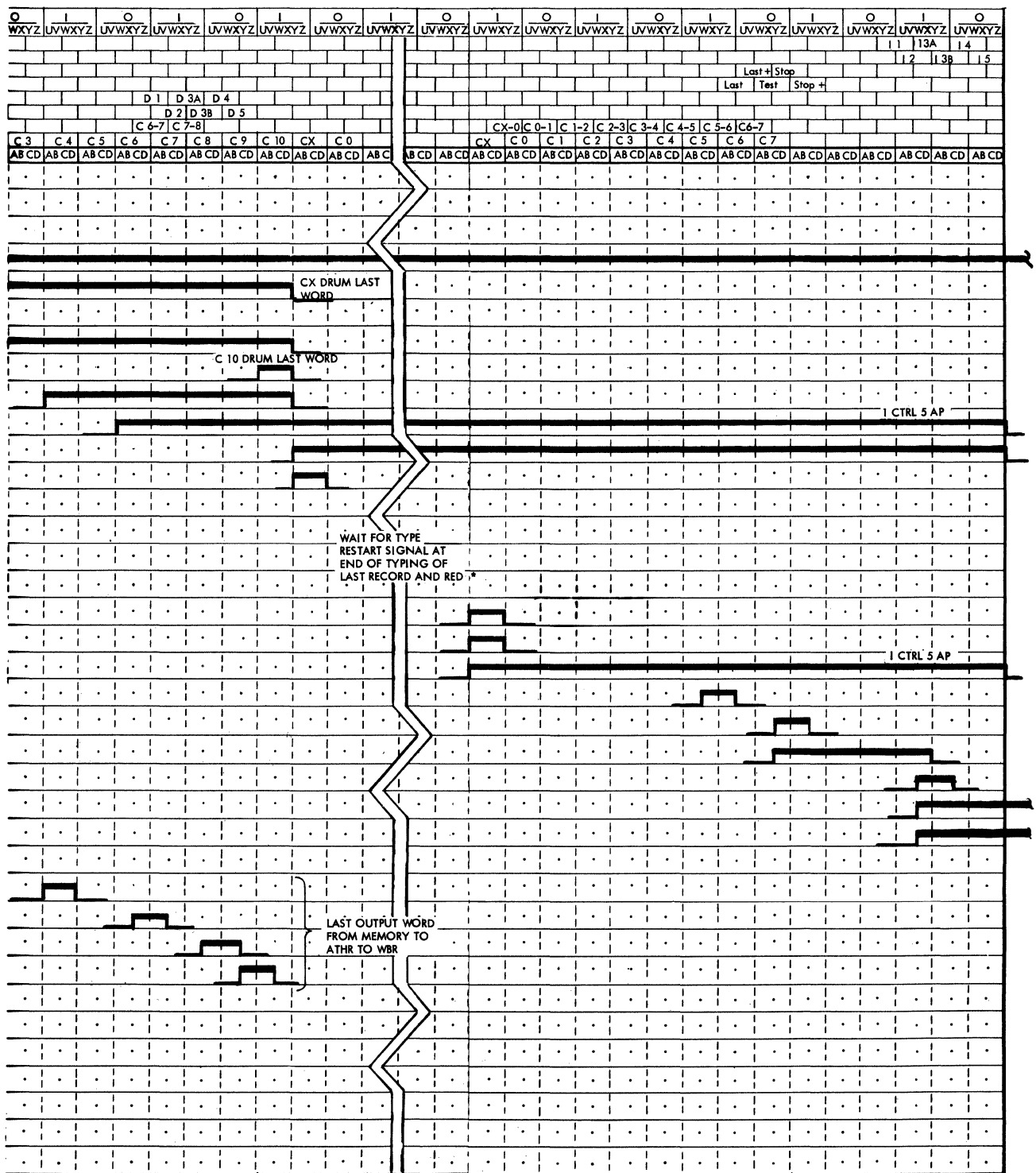


FIGURE 5.12-13B. SEQUENCE FOR RCW ERROR OPERATION

5.13.00 CONSOLE OPERATIONS

5.13.01 Display on Stop

Introduction

When any of the following conditions occur, the 7070 stops and the contents of both the instruction counter and program register are automatically typed out:

- | | |
|----------------------|--------------------------|
| 1. Program Stop | 6. Hang-up Stop |
| 2. Console Stop | 7. Any Error |
| 3. Address Stop | 8. Clocking Error |
| 4. Single Cycle Stop | 9. Address Circuit Error |
| 5. Stop Key | |

Circuits (Figures 5.13-1, 2, 3)

On the detection of Any Error or depression of the Console Stop Key, the instruction being performed is completed. An end-data operation signal results in an instruction cycle. The I Ctrl 3A gate (Figure 5.13-2) then switches with the any-error latch output, or the console-stop signal to turn on the CPU stop latch. The output of the CPU stop latch then turns on PR Inh Ctrl. This gate prevents the following normal operations:

1. Reset of the PR.
2. RO of the IC to the CAB.
3. RI of Program Register from the IB.
4. One-Up of IC

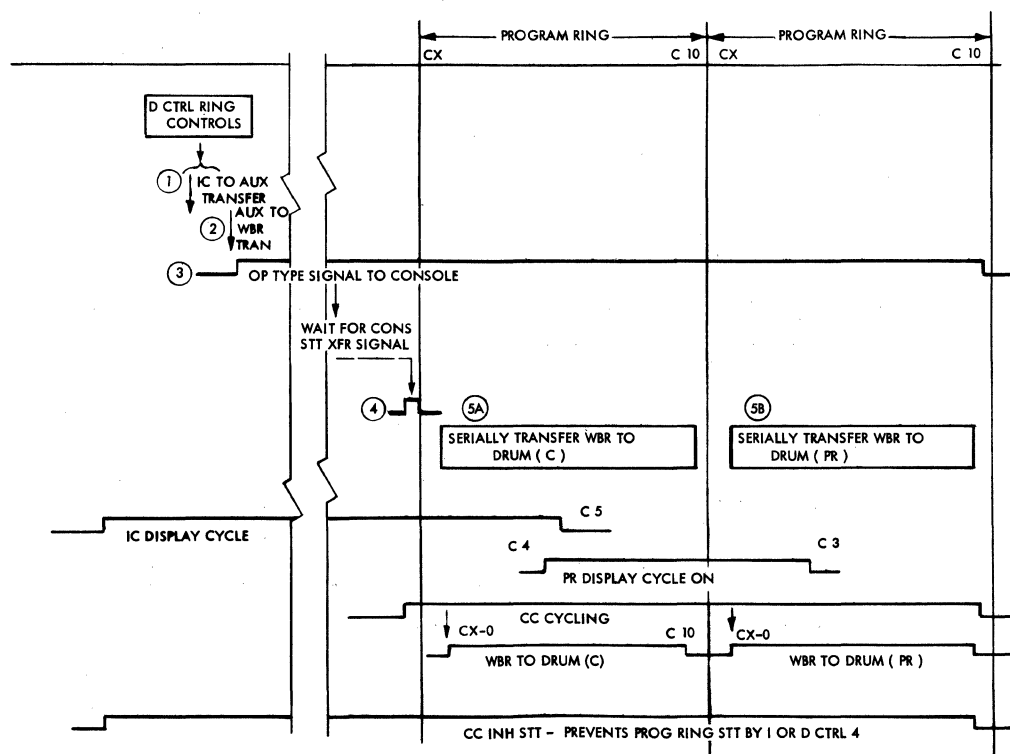


FIGURE 5.13-1. DISPLAY ON STOP OPERATION

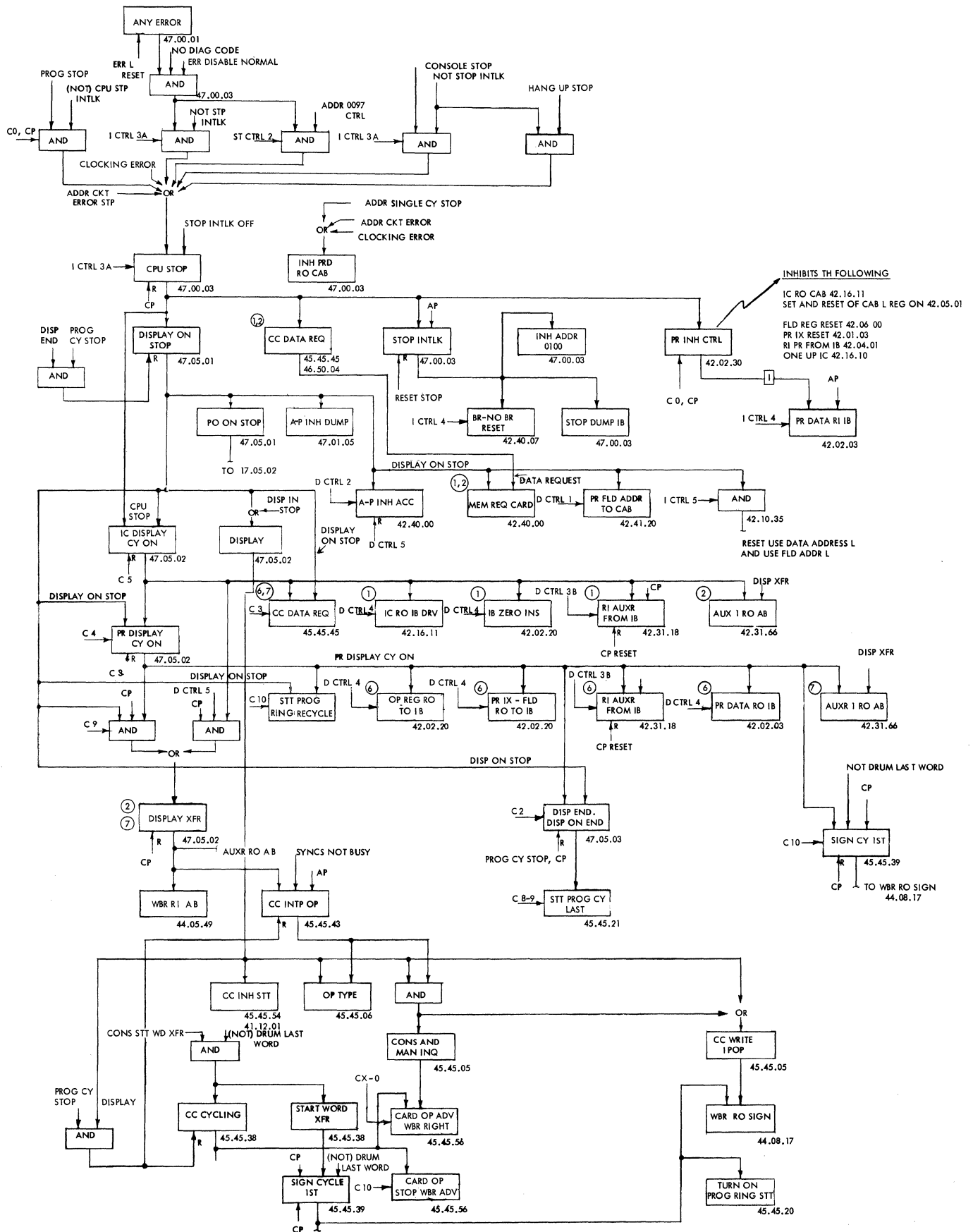


FIGURE 5.13-2. LOGIC FOR DISPLAY ON STOP

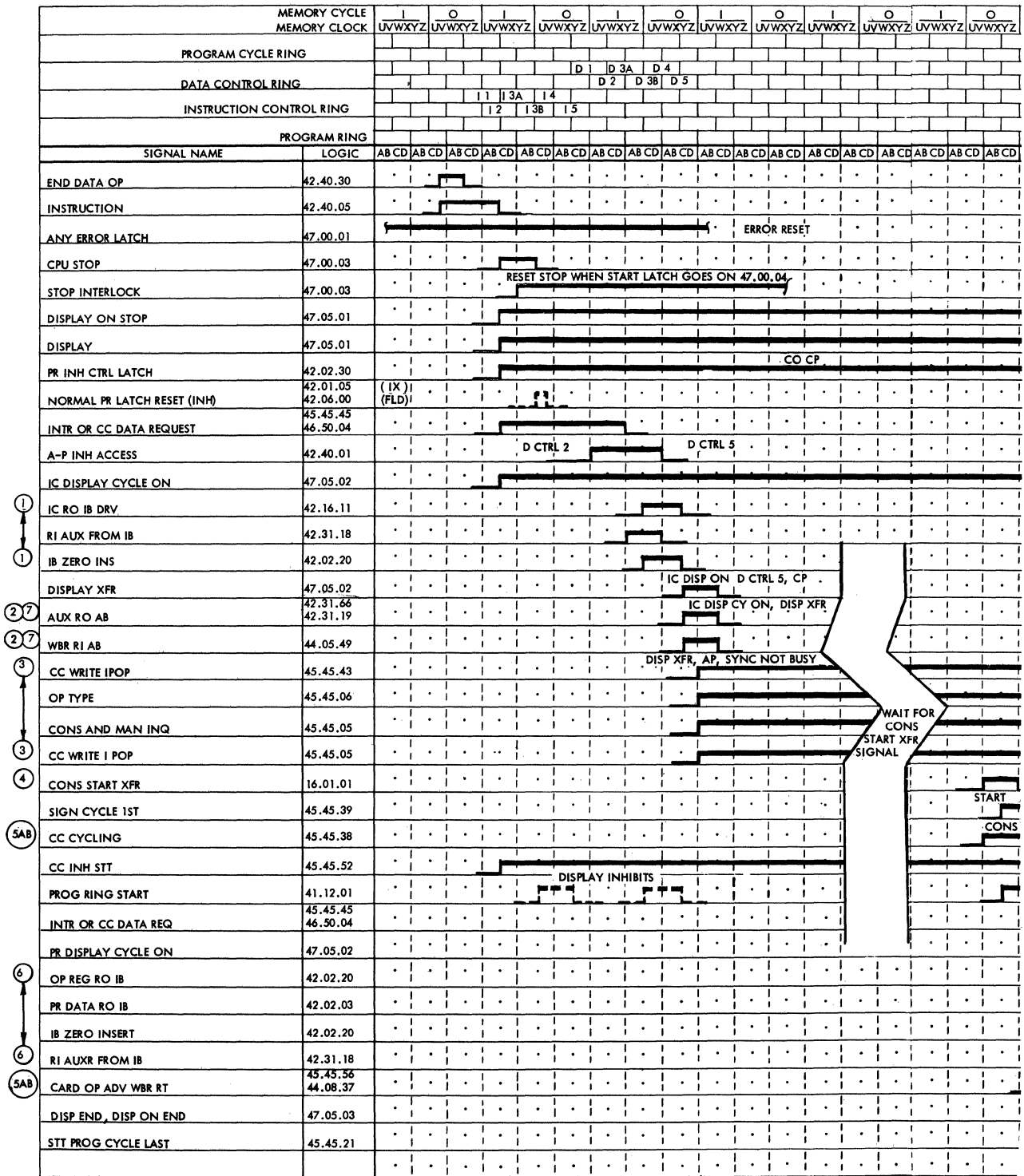


FIGURE 5.13-3A. SEQUENCE FOR DISPLAY ON STOP

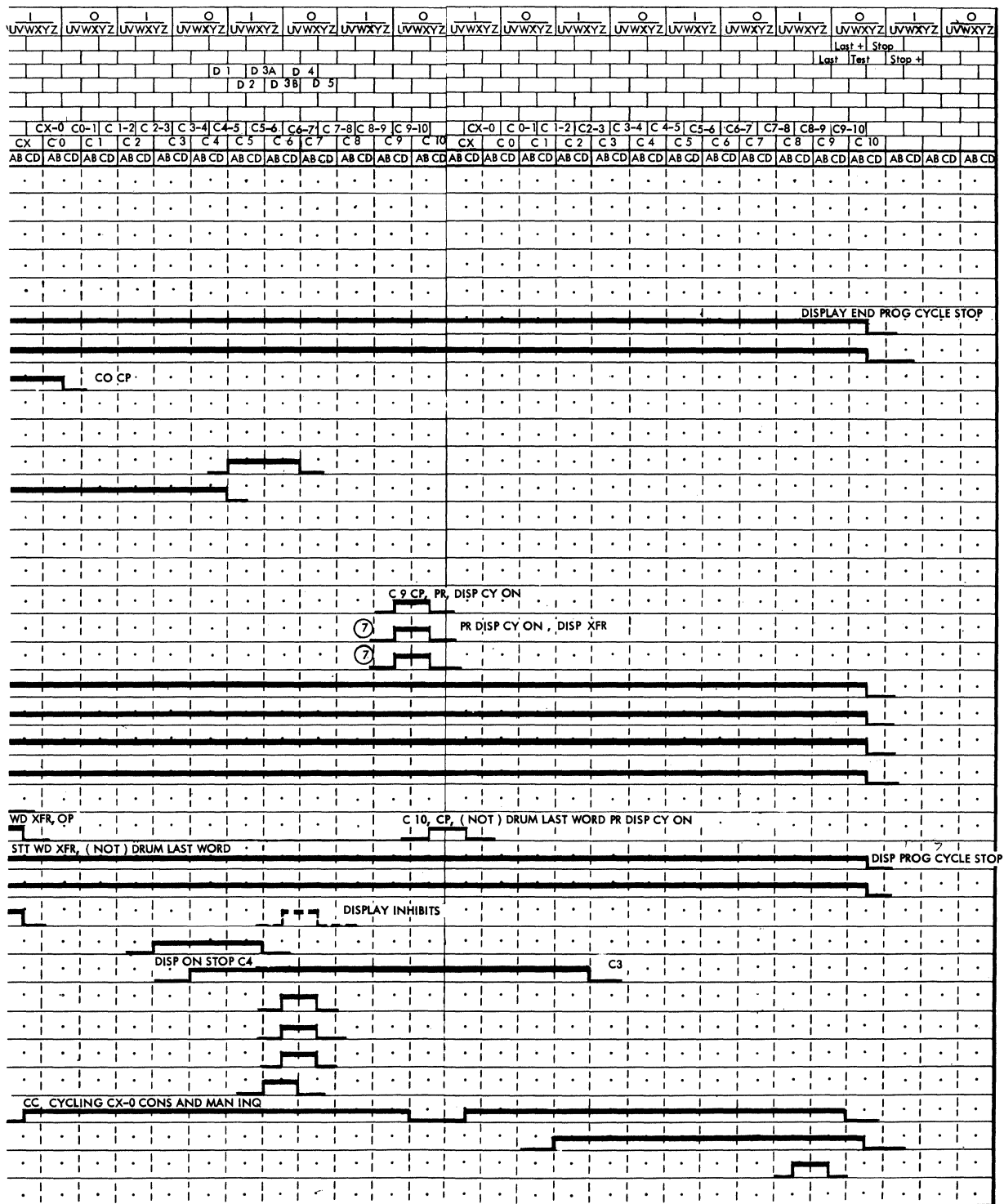


FIGURE 5.13-3B. SEQUENCE FOR DISPLAY ON STOP

Thus, when the 7070 stops, the PR contains the instruction just performed, and the IC contains the address of the next instruction.

The CPU stop signal turns on the display-on stop latch, which controls the entire operation. Circled numbers refer to test and illustrations (Figures 5.13-1, 2, 3). Figure 5.13-1 is a block diagram of the operation. The D control ring is started to control the timing of the parallel-transfers of IC and PR to the auxiliary register. Access is inhibited by A & P Inh Access. The normal program ring start at D Ctrl 4 time is inhibited by CC Inh Start. See Figure 5.13-2 for the positive logic for the entire operation.

① Display-on stop switches with CPU stop to turn on the IC display-cycle latch. This controls the transfer of the IC to the drum buffer. The CPU stop gate had previously caused a data request and the ensuing start of the D Ctrl Ring. At D Ctrl 4 time, the IC is read out to information bus positions 2, 3, 4 and 5. Zeros are inserted in the remaining positions. The auxiliary register reads in from the IB.

② At D Ctrl 5, CP time, the Display Xfr latch is turned on to time the transfer of the auxiliary register to the WBR. In addition, the Display Xfr signal turns on CC Interpret Op, provided the Sync Not Busy line is up. The CC Interpret Op signal switches with Display to develop the following:

- a. ③ the Op Type signal is sent to the console to prepare for the type out of the IC and PR. A Cons Stt Word Xfr signal is returned to the A & P circuits when the drum is ready to receive information from the WBR.
- b. The Cons and Man Inquiry signal is sent to logic 45.45.56 to cause the WBR contents to shift right.

④⑤A The Cons Stt Word Xfr signal turns on CC cycling which, in turn, switches with Cons and Man Inq to cause the serial shift right of the WBR to the drum. The Cons Stt Word Xfr signal also causes the start of the program ring.

At program ring C3 time a data-request operation is initiated. The D Ctrl Ring is started and access is inhibited as on the previous data-request operation.

At program ring C4 time the PR Display Cycle latch is turned on to control the transfer of the program register contents from the PR to the drum buffer.

⑥ PR Display Cycle On switches with D Ctrl 4 to read out the PR to the information bus. The auxiliary register reads in from the IB at D Ctrl 3B, CP time (D Ctrl 4).

⑦ The Display Xfr latch comes on again at C9, CP time under control of PR Display Cycle to time the parallel-transfer of auxiliary register to WBR.

⑤B With the PR contents in WBR, the program ring is recycled at C10 time under control of PR Display Cycle. CC Cycling remains on, while the program ring recycles, to time the serial advance of the WBR to the right.

At C2 time of the recycled program ring, display-end is turned on. This switches with C8-9 to start the program cycle last ring which turns off CC Cycling and CC Intp Op at Prog Cy Stop time. See Figure 5.13-3 for the sequence of the operation.

5.13.02 Display on Console Keyed Address

Introduction

When a core storage, IC, PR, or accumulator address is typed on the console typewriter (address request), the contents of that word are automatically typed (address reply). Section 4.4.01 of the console manual describes the console circuit operation. The typed address enters the address start register as it is typed as described in Section 4.4.01. As a result of the fourth key stroke, the Console Keyed Address End signal (Cons K A End) is sent to the A & P circuits to cause the contents of the location specified by the Addr Start Register to be transferred to the drum buffer via the WBR.

Circuits

Figures 5.13-4, 5 show both logic and sequence for the operation. In general, the procedure is similar to the display-on stop operation previously described except that only ten digits are transferred to the drum buffer instead of fourteen. Thus, the program ring is cycled, CX through C10, only once while the WBR is serially transferred to the drum.

The Console K A End signal (Figure 5.13-4) turns on the display-in stop latch which controls the entire operation. The Cons K A End signal also causes a CC data request with the ensuing start of the data control ring. If the address start register contains the IC, PR, or Accum address (Addr 999X) the A & P Inh Access latch is turned on. The operation takes place in one of the following ways, depending upon the address in the ASR:

- ① IC Address. The IC Display Cycle latch causes the parallel-transfer of the IC contents via the IB to the auxiliary register. Zeros are inserted in positions 0, 1, 6, 7, 8, 9 of the IB.
- ② PR Address. The PR Display Cycle latch causes the transfer of the PR contents to the auxiliary register via the IB.
- ③ Core Address. A normal data-request operation causes the read out of the ASR address to the CAB at D Ctrl 1 time and the read in of data from core storage to the arithmetic register at D Ctrl 4 time.
- ④ Accumulator Address. The designated accumulator is read out via the arithmetic bus to the arithmetic register.
- ⑤ At D Ctrl 5, CP time a Display Xfr signal is developed which causes the transfer of either arithmetic register or auxiliary register to the WBR.
- ⑥ The Display Xfr signal also turns on CC Interpret Op. This results in an Op Type signal to the console. When the console is ready to accept the WBR contents, a Cons Stt Word Xfr signal is returned to the A & P circuits. This results in the turn on of CC Cycling, the start of the program ring, and the serial transfer of WBR contents to the drum. These operations occur in the same manner as previously described in Section 5.13.01.

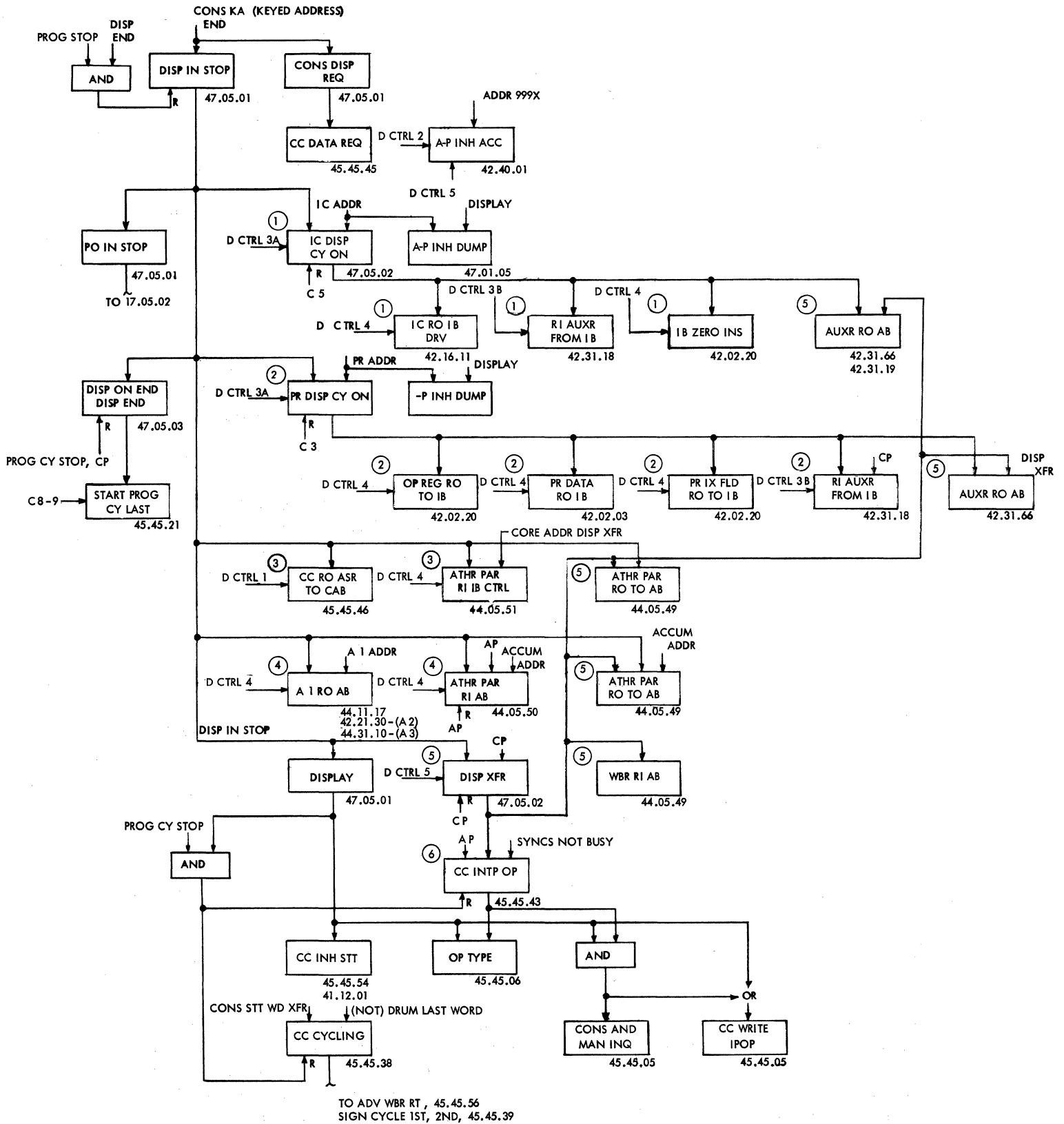
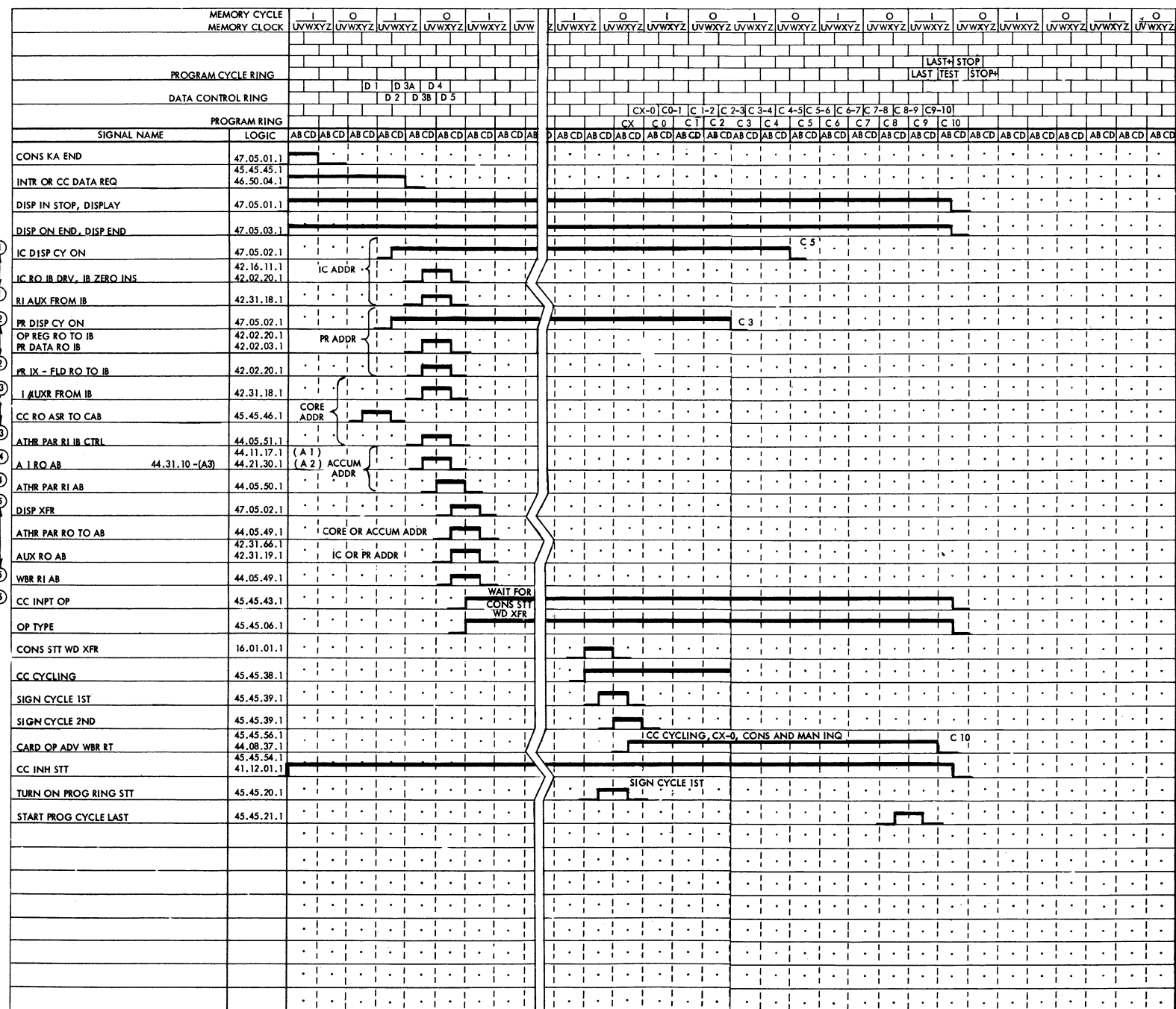


FIGURE 5.13-4. LOGIC FOR DISPLAY ON CONSOLE KEYED ADDRESS



Introduction

Circuits

[illegible]

444

Core Address. The ASR contents are parallel-transferred to core storage at St Ctrl 1 time. The contents of the arithmetic register are then sent to the selected core storage address at St Ctrl 3 time via the IB.

Program Register Address. The PR latches are reset at St Ctrl 2 time and the arithmetic register contents are parallel-transferred to the PR at St Ctrl 3 time.

Accumulator Address. The arithmetic register contents are parallel-transferred via the AB to the designated accumulator at St Ctrl 3 time.

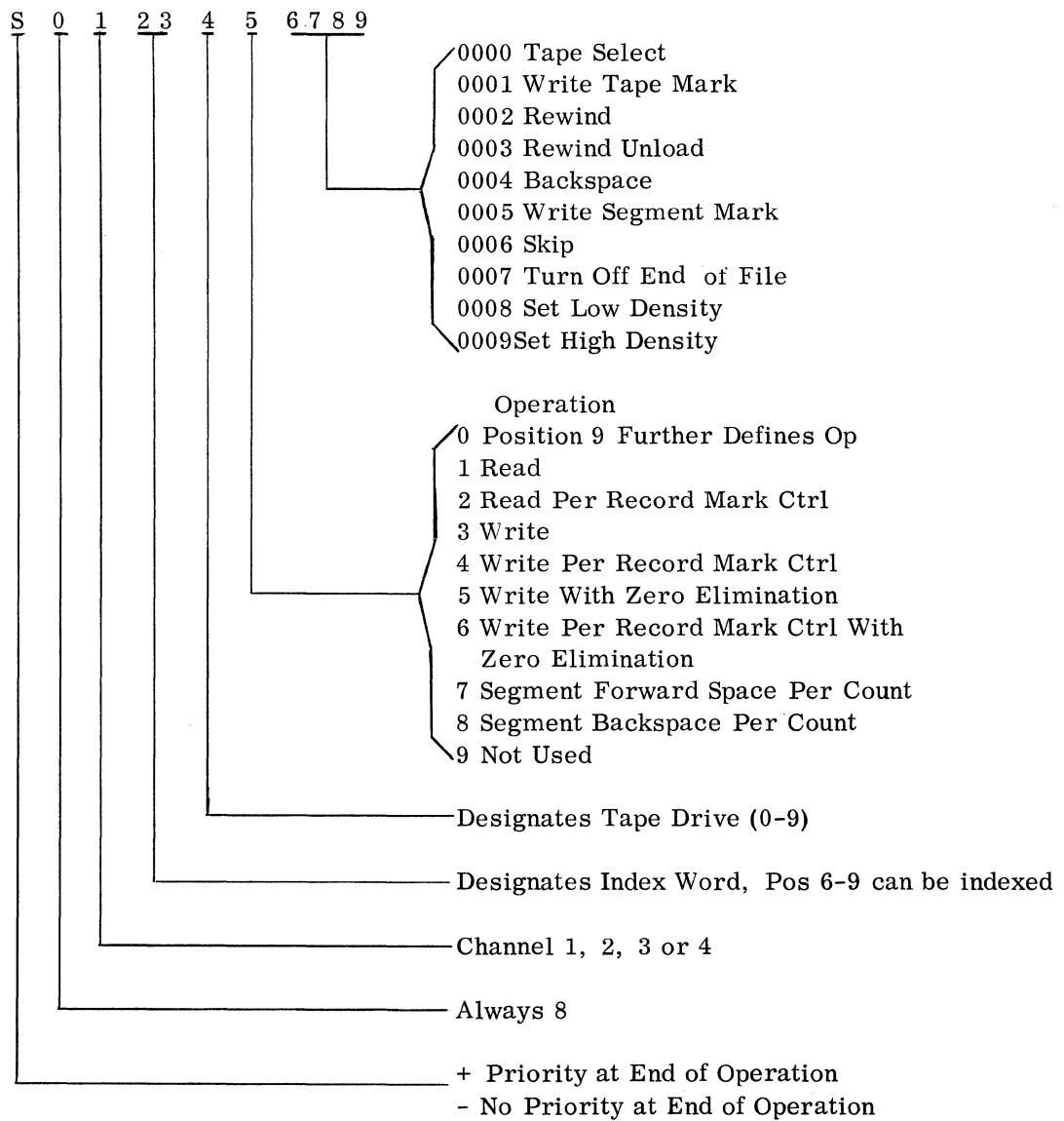
Instruction Counter Address. When the typed address request specifies the IC, an IC Address to console signal (47.01.05) is returned to the console, when the operator types the altered IC value, the console circuits (17.01.02) cause the typed values to enter the ASR instead of the WBR. After the store key is depressed and the Manual Store Op begins, the ASR is parallel-transferred to the IC at St Ctrl 3 time as shown on Figure 5.13-6.

5.14.00 TAPE CHANNEL OPERATIONS

The following explanation deals with the control latches and timings within the A & P Unit that are necessary to initiate the tape and disk storage operations. The data portion of these instructions (pos 6 - 9) do not require the use of core storage and are considered as no access codes.

5.14.01 Tape Control Operation

The tape control operations are specified by four augmented codes ± 81 , 82, 83 and 84. The units position of the 80's op code specifies the tape channel that is to be used. The actual tape operation is specified by position 5 of the tape instruction.



The explanation of these various tape operations can be found in the 7604 Tape Control manual (Volume III).

Before a tape operation can be initiated, two tests are performed at the beginning of the tape instruction. First, a check is made to insure that the addressed channel is not busy. With the addressed channel busy, the A & P unit waits until the channel is finished with the previous operation. Under this condition the program-advance light on the 7070 console is flashing to indicate that the program did not advance.

Secondly, a check is made on the stacking latch that is reserved for the addressed channel and tape drive. The stacking latch on, indicates that a previous operation has set the stacking latch and the interrupt has not been serviced. Under this condition the A & P unit does not advance to the next instruction but continues to try to initiate the tape operation.

To perform this test, a store cycle is initiated which uses the normal store controls. The only exception is that the Memory RI is inhibited.

With the channel available, a tape call is sent to the addressed channel and the store request cycle is initiated (Figure 5.14.1). During the store cycle, the channel controls are reset and the contents of the Op Reg, Prog Reg D, and instruction counter are transferred via the information bus to the Buffer B Register in the 7604.

The field register contents (pos 4 and 5 of instruction) are sampled in the 7604 unit to condition Op Register and Select Register latches. The output of the select-register latches is sent to the 7602 where it is switched against the output of the reserved stacking latch. If the stacking latch is on, a tape reply signal is sent to the A & P.

At store 6 time the test is performed to see if a tape reply has been developed. With a tape reply sensed, an end-data op signal is developed and the A & P goes into an instruction cycle to try tape instruction again.

With the stacking latch off, the tape reply signal is not developed and at store 6 time a Start Prog Ring signal is developed. With the program ring running the Instruction Counter is 1-upped and a Continue Sync Op signal is developed which allows the tape channel to be interlocked during the second store cycle.

During the second store cycle the contents of the Op Reg, Prog Reg D and instruction counter are again transferred to Buffer B via the information bus. On this transfer of information the addressed channel sends a restart signal to A & P which allows the machine to move to the next instruction.

With the information transferred to Buffer B, the tape operation can operate independently from the A & P.

Figure 5.14-2 shows the sequence chart for the operation.

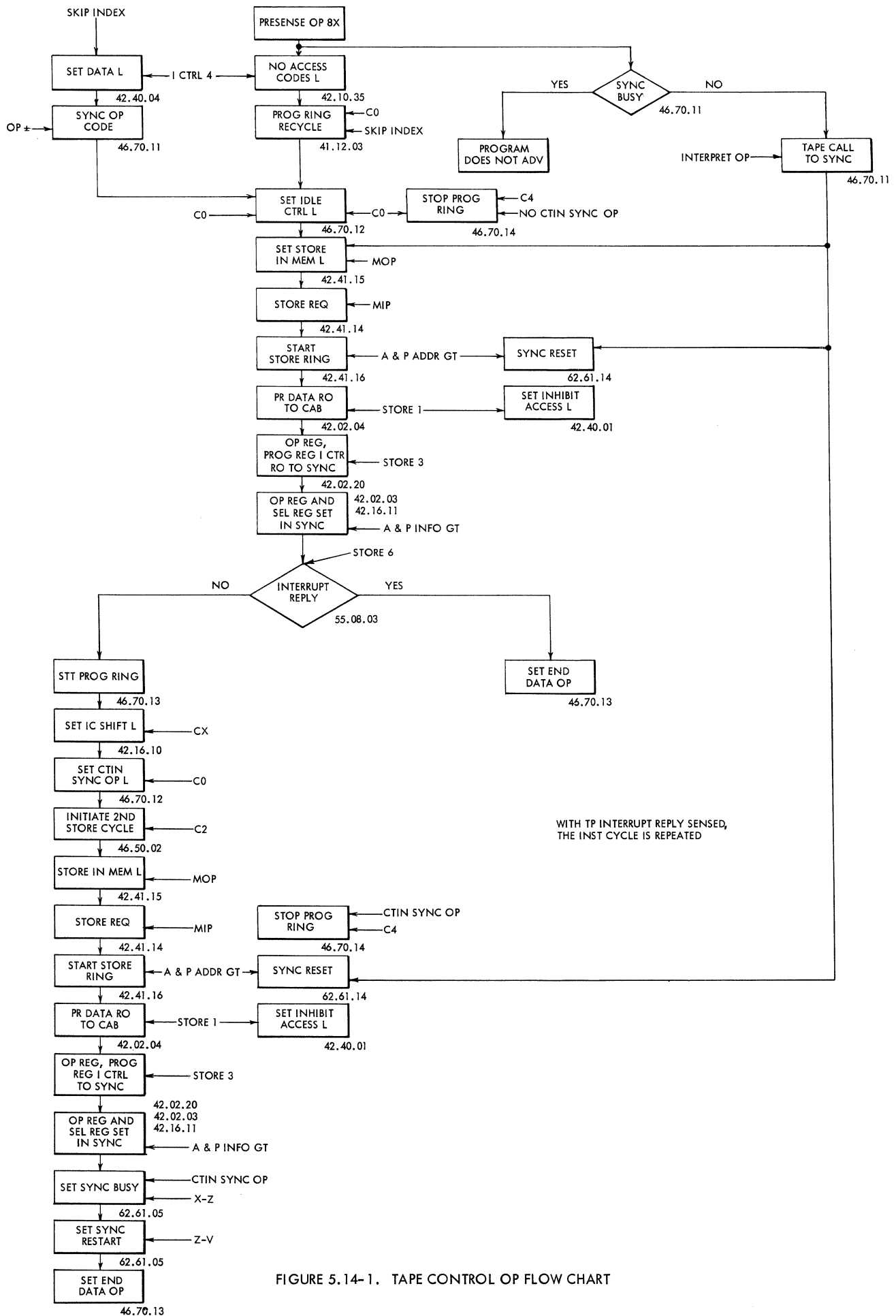


FIGURE 5.14-1. TAPE CONTROL OP FLOW CHART

Sense Tape Operation

With the contents of the instruction word placed onto the information bus a pre-sense 8X signal is developed. At I Ctrl 4 time, and the 8X code, the no access codes latch is turned on to inhibit the data word request. At the same time the contents of positions 2 and 3 of the instruction are analyzed for an index operation. For this explanation assume a no-index operation. With a no-index operation, the data latch is turned on which in turn develops a Sync Op Code signal.

With the 8X signal developed, a test is performed to see that the sync is not busy. It can be seen on Figure 5.14-1 that if the sync is available a tape call is developed when the Interpret Op Code signal is developed.

The tape call signal is used to condition the sync and develop the Tp or DS Store in Memory signal.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set No Access L	8X, I Ctrl 4	2C-42.10.35
Set Data Latch	Skip Index	3D-42.40.04
Develop Sync Op Code	81 or 82	3B or 3C-46.70.11
Recycle Prog Ring	No Access Codes, Skip IX Cy, C0	2G-41.12.03
Set Idle Ctrl	Sync Op, Plus, or Minus, C0	2A-46.70.12
Set Interpret Op Code	Data L, Prog Ring Rcy, DP	3B-42.40.08
Develop Tape Call	Sync Not Busy, Interpret Op, 81	6A-46.70.11
Develop TP DS Store in Mem	Tape Call Idle Ctrl	3H-46.70.14
Stop Prog Ring	No Ctin Sync Op Tape Call, C2	4A-46.70.14

Perform First Store Cycle

With a Tp Store in Memory signal available, the store-in-memory latch is turned on with the following MOP. A Store Req signal is developed the following MIP time and sent to the 7602 unit. If core storage can accept the A & P request, it replies in the form of an A & P Addr and Infor Gate.

At Addr Gate Time, the store ring is started and the addressed sync is reset. At Store Ctrl 1 Time the contents of Program Reg D are read out to the CAB to condition the address generator. However, because this operation does not require the use of core storage the Inhibit Access latch is turned on.

At store ctrl 3 time the contents of the op register, Prog Reg D, and Inst Counter are read out to the information bus.

At info gate time, the contents of the information bus is read into Buffer B. The static output of the field register positions is used at this time to condition the op register and select register. The op register output is used to determine the tape operation. The select register output is used to determine the tape drive that is to be used.

The selected tape drive is switched with the static output of the stacking latch in the 7602 to determine if an Interrupt is waiting for the addressed sync and tape drive. With the stacking latch on, a Tp Code Reply signal is developed and sent to A & P.

At Store ctrl 6 time, the test is made to determine whether the instruction is to continue or is to be repeated. For this explanation assume that the stacking latch was off. With this condition, a Start Prog Ring signal is developed and the program ring turns on.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Store in Mem	Tape DS St Mem, MOP	3C-42.41.15
Set Store Req	Store in Mem, MIP	2B-42.40.14
Start Store Ring	Store Req, A & P Addr Gt	4C-42.41.16
Reset Addressed Sync	Tape Call, A & P Addr Gt	4D-62.61.14
RO PR to CAB	Op Sync Codes, Store 1	4H-42.02.04
Set Inhibit Access L	Sync Op Codes, Store 1	4F-46.70.14
RO Op Reg to IB	Op Sync Codes, Store 3	4E-42.02.20
RO Prog Reg to IB	Op Sync Codes, Store 3	2G-42.02.03
RO Inst Counter to IB	Op Sync Codes, Store 3	4E-42.16.11
Set Op Register	Sample Gate	62.61.09
Set Sel Register	Field Reg Sel Spl Fld Reg Output	62.61.04

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Test for TP Reply	Stacking Latch and Tape Drive	55.08.03
Develop Prog Ring Stt	No Tp Reply, Tape Call, St 6	4F-46.70.13
Develop End Data Op	Tp Reply, Store 6	3D-46.70.13

Perform Second Store Cycle

With the program ring on, the instruction-counter shift-left latch is turned on at CX time to increase the contents of the instruction counter by one. Because a no-tape reply has been sensed, a Ctin Sync Op is developed at C0 time and sent to the sync to condition the turn on of the Busy latch.

At C2 time the Ctin Sync Op signal is used to initiate the second-store cycle. The second-store cycle is performed in the same manner as the first. At addr gate time the sync is reset. At information gate time, the Op Reg, Inst Counter and Program Register D are again transferred via the information bus to Buffer B. With a Ctin Sync Op available a restart is developed and sent to the A & P. This signal develops an end-data op signal which allows the program to advance to its next program step.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Stt Prog Ring	No Tp Reply, St 6 Tape Call	4F-46.70.13
Set IC Sh Left L	Start 1 up, AP	3D-42.16.10
Set Ctin Sync Op	IC Sh L, CX-0, Tape Call, AP	4F-46.70.12
Initiate 2nd Store Cycle	Tp Call, Ctin Sync Op C 2	2G-46.50.02
Set Store in Mem	Store in Mem	5F-42.41.15
Set Store Req	Store in Mem, MIP	2B-42.41.14
Start Store Ring	Store Req A & P Addr Gt	4C-42.41.16
Reset Addr Sync	Tape Call A & P Addr Gt	4D-62.61.14
RO PR to CAB	Op Sync Codes, Store 1	4H-42.02.04

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Inhibit Access L	Op Sync Codes, Store 1	4F-46.70.14
RO Op Reg to IB	Op Sync Codes, Store 3	2G-42.02.03
RO Prog Reg RO to IB	Op Sync Codes, Store 3	2G-42.02.03
RO Inst Counter to IB	Op Sync Codes, Store 3	4E-42.16.11
Set Op Register	Sample Gate	62.61.09
Set Sel Register	Fld Reg Sel Spl Fld Reg Output	62.61.04
Set Sync Busy	Ctin Sync Op X-Z, Restart Timer	2G-62.61.05
Set Restart Timer	Sample Gate, V-X	3A-62.61.05
Set Restart	Busy-Restart Timer, Z-V	2D-62.61.05
Set End Data Op	Tape Op Tp Restart	3E-46.20.13

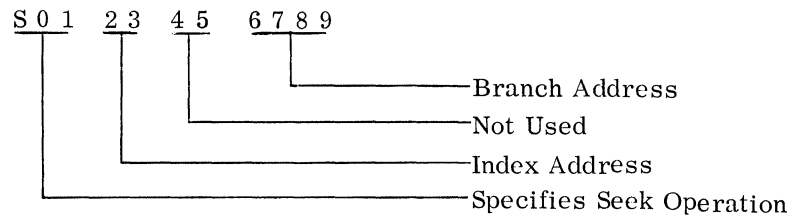
5.14.02 Disk Seek Operation

The + 95 DS Seek instruction is used with the contents of accumulator 3 to determine the box, arm, disk and track location.

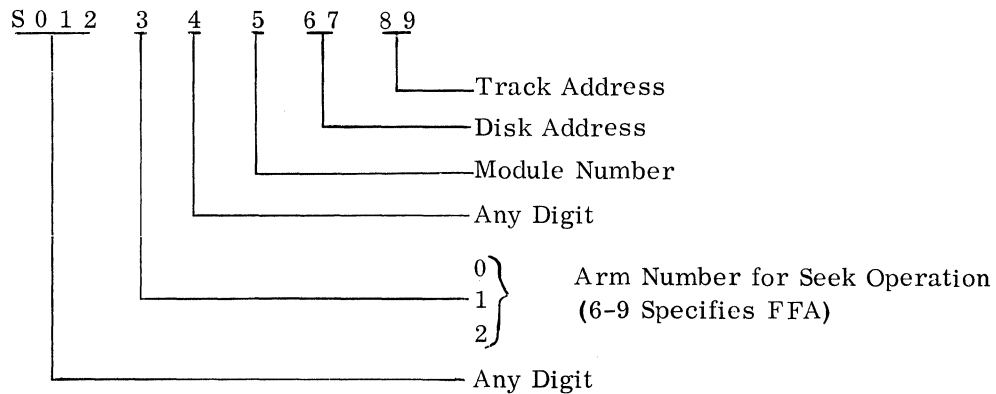
Two types of seek operations can be performed. One, a seek operation using a specified arm. If the specified arm is busy, the next program step is taken from the data portion of the seek instruction. Secondly, a find-free-access operation, which automatically finds a free arm in the addressed file unit, then proceeds to seek to the address specified by positions 6 - 9 of accumulator 3. With no arm available at the completion of a FFA operation, a digit 9 is inserted into position 3 of accumulator 3. Under this condition the next instruction is taken from the data portion of the + 95 instruction.

The contents of accumulator 3 and the + 95 seek instruction is as follows:

Instruction Word



Accumulator #3



Sense Seek Operation

With the contents of the instruction word placed onto the information bus a pre-sense 9X signal is developed. At I Ctrl 4 time and a 9X code the no-access-codes latch is turned on to inhibit the data word request. At the same time, the contents of positions 2 and 3 of the instruction are analyzed for an index operation. With a no-index operation, the data latch is turned on which in turn develops the Sync Op Code signal.

It can be seen on Figure 5.14-3 that with the Interpret Op Code signal available the + 95 Seek code is sensed. With the + 95 code, a DS Control signal is developed which is used to turn on the DS store-in-memory signal.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set No Access L	9X	4J-42.10.20
Set Data Latch	Skip Index	3D-42.40.04
Develop Sync Op Code	DS Control Codes Op + or -	6E-46.70.11
Recycle Prog Ring	No Access Codes Skip IX Cy, C0	2G-41.12.03

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Busy Idle Ctrl	Sync Op, Plus or Minus, C0	2A-46.70.12
Set Interpret Op Code	Data L, Prog Ring Rcy, DP	3B-42.40.08
Develop + 95	Ctin Sync Op, Op + 9X, X 5	3F-46.70.11
Develop Tp DS Store in Mem	DS Control Code Matrix Not Busy Busy Idle Ctrl.	2H-46.70.14

Transfer A3 to Arithmetic Register

To perform the seek operation the contents of Accumulator 3 must be parallel-transferred via the arithmetic bus to the arithmetic register. To accomplish this a store cycle is generated to furnish the necessary timing to develop the A3 RO to AB signal and also condition the Arith Reg Par RI Ctrls.

With a DS store-in-memory signal available, the store-in-memory latch is turned on with the following MOP pulse. A Store Req signal is developed the following MIP time and sent to the 7602 unit. If memory can accept the A & P request, it replies in the form of any Addr and Info Gate.

At addr gate time the store ring is started. At store ctrl 1 time the contents of program register D are read out to the CAB to condition the address generator. However, because this operation does not require the use of core storage the inhibit-access latch is turned on.

During store ctrl 1 AP time the A3 RO to AB signal is brought up to transfer the contents of A3 to the arithmetic bus. The following CP time the arithmetic register Par RI controls are conditioned to read in from the bus.

With the contents of A3 transferred to the arithmetic register, the program ring is started. The program ring furnishes the necessary timing for serially reading the contents of the arithmetic register to channel 2 for validity checking. At the same time the arithmetic register contents are sent to the 7605 disk storage control unit.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Store in Mem L	DS St in Mem MOP	3C-42.41.15
Set Store Req	Store in Mem MIP	2B-42.41.14
Start Store Ring	Store Req A & P Addr Gt	4C-42.41.16
RO PR to CAB	Op Sync Codes, Store 1	4H-42.02.04

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Inhibit Access L	Op Sync Codes, Store 1	4F-46.70.14
Start Prog Ring	DS Control Codes, Store 1	3B-41.12.01
A 3 RO to AB	DS Control Codes, Store 1, AP	3J-44.31.10
Arith Reg Par RI from AB	DS Control Codes, Store 1, CP	5H-44.05.54

Serially RO Arithmetic Register

Figures 5.14-3 and 5.14-4 show that the digit positions of the arith reg are selected at random intervals. This operation is accomplished by gating the field register controls which allow the specified digit position to be read out to the output latches at a specific program ring timing. The output of the arithmetic register is then gated to channel 2 for validity checking and also sent to the 7605 unit for arm, disk and track selection.

At C0 time, the true-add latch is turned on to inhibit the channel 2 zeros. Channel 1 zeros are inserted during this operation. During this time the Martix RI latch is conditioned in the 7605.

At C1 time the arithmetic register 5th position is read out to channel 2 and to the DS units. This position contains the module number that determines which file unit is to be used. Module number assignments are discussed in the disk storage section of the 7070 Data Processing System Reference Manual Form A24-7003.

At C2 time, position 3 is read out to the 7605. This position selects the arm that is to be used for the seek operation. With a 0, 1 or 2 the specified arm is selected. However, if a 3 through 9 is sensed a find-free-access latch is turned on in the 7605 and a free arm is found.

At C3 time, the module number is gated back to the arithmetic register via the adder. Due to the digit delay through the adder the module number is inserted into position 4 at C4 time.

At C4 time the arm number is also sent back via the adder to the arithmetic register. This information is placed into position 3 at C5 time.

At C5 and C6 time, the track address is read out to the disk unit. The following program ring timings (C7 and C8) the disk address is sent to the disk unit. With both the track and disk address read out to the 7605, the selected arm is moved to the specified address. At C9 and C10 time both the module and arm number are again transferred to the 7605.

With the transfer of data completed, both the Arithmetic Par RO to AB and the Accumulator 3 AB RI controls are brought up to transfer the arithmetic register contents back to accumulator 3.

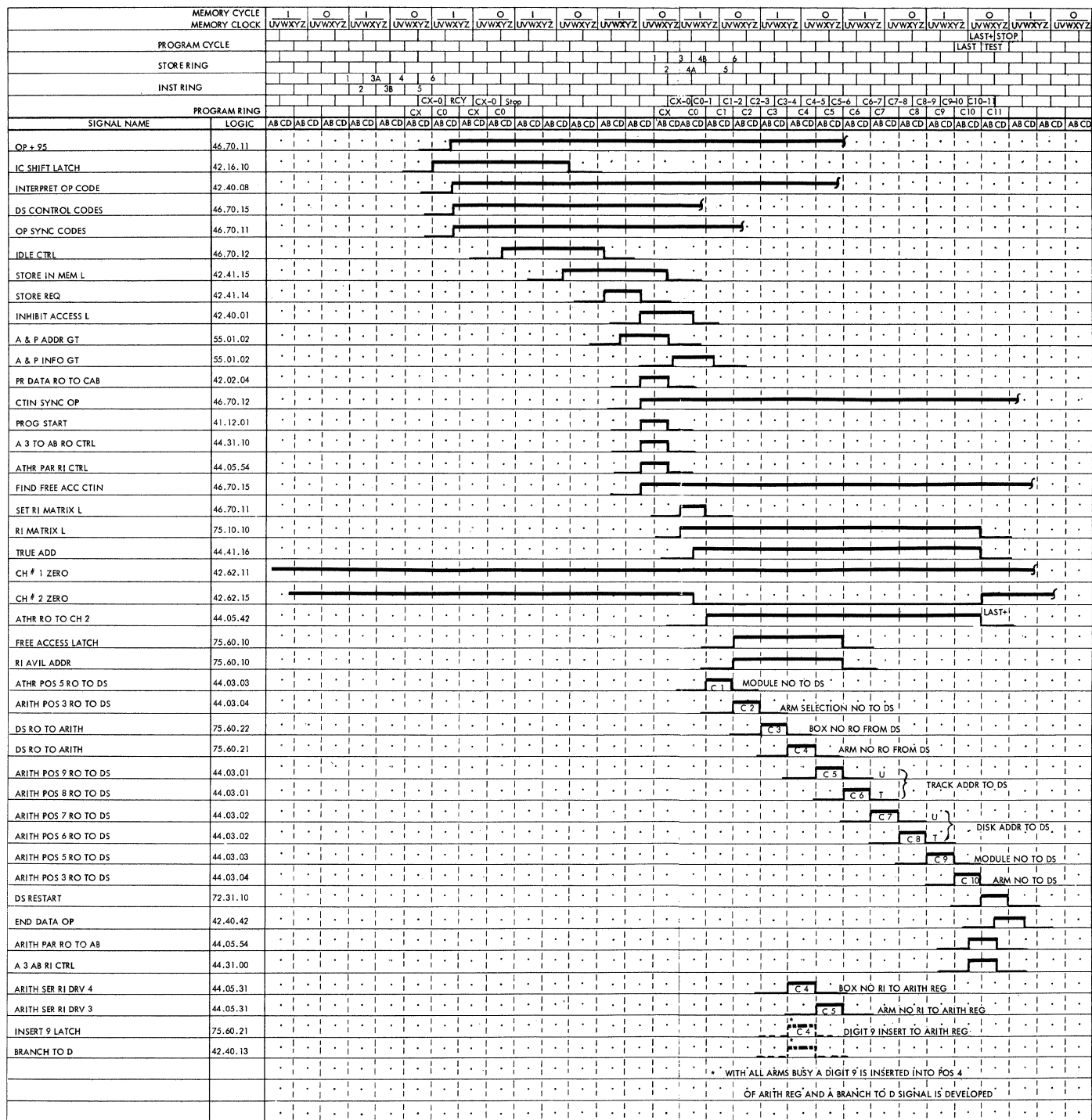


FIGURE 5.14-4. DISK SEEK OPERATION

With the transfer completed, a DS Restart signal is sent to the A & P unit. This signal develops an end-data op signal which allows the program to advance to the next instruction specified by the instruction counter.

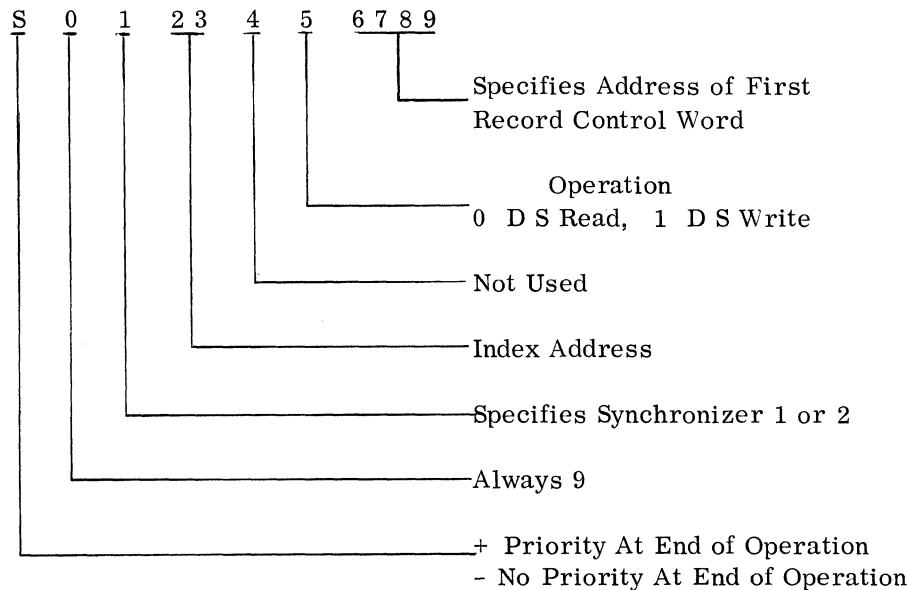
If the arm was busy or no access arm was available, the next instruction would be specified by the contents of the Program Register D.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Start Prog Ring	DS Codes Stt Ctrl, Store 1	3B-41.12.01
Set Matrix RI L	Find Free Acc Ctin C0	3H-46.70.11
Set True Add	Find Free Acc Ctin C0-1	4B-44.41.16
Set Ch # 1 Zeros	Find Free Acc Ctin C 0-1	2G-42.62.11
Arith Pos 5 RO	F 6-5, AP	4C-44.03.03
Arith Pos 3 RO	F 4-3, AP	4C-44.03.04
Arith Ser RI Drv 4	Op FFA + 95 C4	5B-44.05.31
Arith Ser RI Drv 3	DS Insert Ctrl, Find Free Acc Ctin, C5	5D-44.05.31
Arith Pos 9 RO	F 1-9, AP	4C-44.03.01
Arith Pos 8 RO	F 9-8, AP	4H-44.03.01
Arith Pos 7 RO	F 8-7, AP	4C-44.03.02
Arith Pos 6 RO	F 7-6, AP	4H-44.03.02
Arith Pos 5 RO	F 6-5, AP	4C-44.03.03
Arith Pos 3 RO	F 4-3, AP	4C-44.03.04
Arith Par RO to AB	Find Free Acc Ctin, C 10-11	5C-44.05.54
A 3 AB RI Ctrl	Find Free Acc Ctin, C 10-11	5J-44.31.00
Develop DS Restart	Any DSUL On, C 11 + 95	6A-75.60.11
Set End Data Op	Find Free Acc Ctin DS Restart, CP	3F-42.40.41

5.14.03 Disk Read/Write Operation

The disk read/write operation is performed in much the same way as a tape read or write operation. The disk read/write operation is specified by two augmented op codes ± 91 or 92 .

The format for a disk read/write operation is as follows:



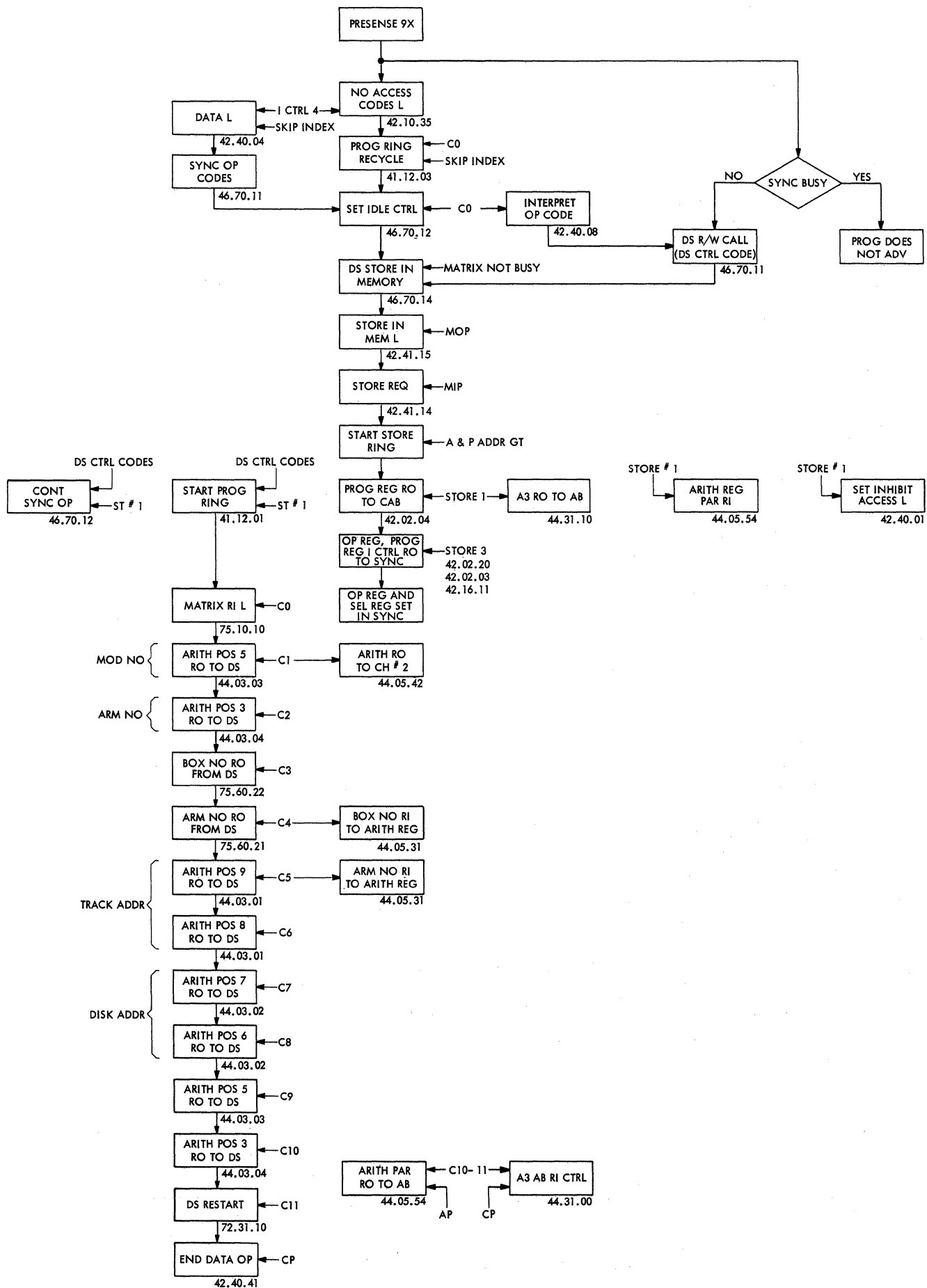
The disk read/write instruction is used with the contents of accumulator 3. The contents of accumulator 3 are transferred to the arithmetic register where it is serially transferred to the 7605. The contents of accumulator 3 are used to compare the arm, track and disk address which has been set by the seek operation. If the disk, track, or arm addresses do not compare, a program address check error (condition 6) is developed and FSW cycle is initiated and the disk read/write operation is discontinued.

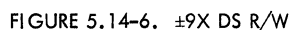
The final status word is assembled in the 7604 unit and stored into its prescribed core storage location.

Prior to the transfer of the arithmetic register, the availability of the address channel is checked (Figures 5.14-5 and 5.14-6). With the channel available, a DS read/write operation is developed which initiates a store cycle. The store cycle furnishes the necessary timing to transfer the contents of the Op register, instruction counter, and the Program Register D to Buffer B in the 7604.

Sense Disk Read/Write Operation

With the contents of the instruction word placed onto the information bus, a pre-sense 9X signal is developed. At I Ctrl 4 time, and the 9X code, the no-access-codes latch is turned on to inhibit the data-word request. At the same time the contents of the positions 2 and 3 of the instruction are analyzed for an index operation. For this explanation assume a no-index operation. With a no-index operation, the data latch is turned on which in turn develops a Sync Op Code signal.





With the 9X signal developed, a test is performed to see that the sync is not busy. Figure 5.14-5 shows that if the sync is available, a disk read/write call is developed when the Interpret Op Code signal is developed.

With a disk read/write call developed, a DS store-in-memory signal is initiated to condition the store-in-memory latch.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set No Access L	9X	4J-42.10.20
Set Data Latch	Skip Index	3D-42.40.04
Develop Sync Op Code	DS Control Code Op + or -	6E-46.70.11
Recycle Prog Ring	No Access Code Skip IX Cy, C0	2G-41.12.03
Set Idle Ctrl	Sync Op, Plus or Minus, C0	2A-46.70.12
Set Interpret Op Code	Data L, Prog Ring Rcy, DP	3B-42.40.08
Develop Disk R/W Call	Sync Not Busy Interpret Op, 9X	6A-46.70.11
Develop TP DS Store in Mem	Tape Call, Idle Ctrl	3H-46.70.14
Stop Prog Ring	No Ctin Sync Op, Tape Call, C2	4A-46.70.14

Initiate Store Cycle

With a disk store-in-memory signal available, the store-in-memory latch is turned on with the following MOP. A store-request signal is then developed the following MIP time and sent to the 7602. If core storage can accept the A & P request, it replies in the form of the Addr and Info Gate.

At addr gate time the store ring is started and the addressed sync is reset. At store ctrl 1 time, the contents of the program reg D are read out to the CAB to condition the address generator. However, because this operation does not require the use of core storage, the inhibit-access latch is turned on.

During store ctrl 1 AP time, the A3 RO to AB signal is brought up to transfer the contents of A3 to the arithmetic bus. The following CP time the arithmetic register par RI ctrls are conditioned to RI from the arithmetic bus. At store 1 time the Continue Sync Op signal is developed and sent to the 7604 unit to interlock the addressed channel.

With the contents of A3 transferred to the arithmetic register, the program ring is started. The program ring furnishes the necessary timing for serially reading the contents of the arithmetic register to channel 2 for validity checking. At the same time the arithmetic register contents are sent to the 7605 disk storage control unit

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Store in Mem L	DS St in Mem, MOP	3C-42.41.15
Set Store Req	Store in Mem, MIP	2B-42.41.14
Start Store Ring	Store Req, A & P Addr Gt	4C-42.41.16
Reset Addressed Sync	DS R/W Call A & P Addr Gt	4D-62.61.14
RO PR to CAB	Op Syn Codes Store 1	4H-42.02.04
Set Inhibit Access L	Op Sync Codes Store 1	4F-46.70.14
Start Prog Ring	DS Control Codes Store 1	3B-41.12.01
Set Cont Sync Op	DS Control Codes Store 1	4G-46.70.12
A 3 RO to AB	DS Control Codes Store 1 AP	3J-44.31.10
Arith Reg Par RI from AB	DS Control Codes Store 1, CP	5H-44.05.54
RO Op Reg to IB	Op Sync Codes Store 3	4E-42.02.20
RO Prog Reg to IB	Op Sync Codes Store 3	2G-42.02.03
RO Inst Counter to IB	Op Sync Codes Store 3	4E-42.16.11

Serially RO Arithmetic Reg

Figures 5.14-5 and 5.14-6 show that the digit positions of the arithmetic register are selected at random intervals. This operation is accomplished by gating the field register controls which allow the specified digit position to be read out to the output latches at a specific prog ring timing. The output of the arithmetic register is then gated to channel 2 for validity checking and also sent to the 7605 unit for arm, disk, and track selection.

At C0 time, the true-add latch is turned on to inhibit the channel 2 zeros. Channel 1 zeros are inserted during this operation. Also at C0 time the Matrix RI latch is conditioned in the 7605

At C1 time the arithmetic register 5th position is read out to channel 2 and to the disk units. This position contains the module number that determines which file unit is to be used. Module number assignments are discussed in the disk-storage section of the 7070 Data Processing System Reference Manual form A24-7003.

At C2 time, position 3 is read out to the 7605. This position selects the arm that is to be used.

At C3 time, the module number is gated back to the arithmetic register, via the adder. Due to the digit delay through the adder the module number is inserted into position 4 at C4 time.

At C4 time, the arm number is also sent back via the adder to the arithmetic register. This information is placed into position 3 at C5 time due to the digit delay through the adder.

At C5 and C6 time the track address is read out to the disk unit. The following program ring timings (C7 and C8) the disk address is sent to the disk unit.

At C9 and C10 time both the module and arm number are again transferred to the 7605.

With the transfer of data completed, both the arithmetic par RO to AB and the Accumulator 3 AB RI controls are brought up to transfer the arithmetic register contents back to Accumulator 3.

If during the disk locafier check operation a comparison of address is not met, a disk recycle operation is performed.

With the transfer completed a DS restart signal is sent to the A & P unit. This signal develops an end-data op signal which allows the program to advance to the next instruction specified by the instruction counter.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Start Prog Ring	DS Codes Stt Ctrl, Store 1	3B-41.12.01
Set Matrix RI I	Find Free Acc Ctin, C0	3H-46.70.11
Set True Add	Find Free Acc Ctin, C0-1	4B-44.41.16
Set Ch # 1 Zeros	Find Free Acc Ctin, C0-1	2G-42.62.11
Arith Pos 5 RO	F 6-5, AP	4C-44.03.03
Arith Pos 3 RO	F 4-3, AP	4C-44.03.04
Arith Ser RI Drv 4	Op FFA + 95 C4	5B-44.05.31

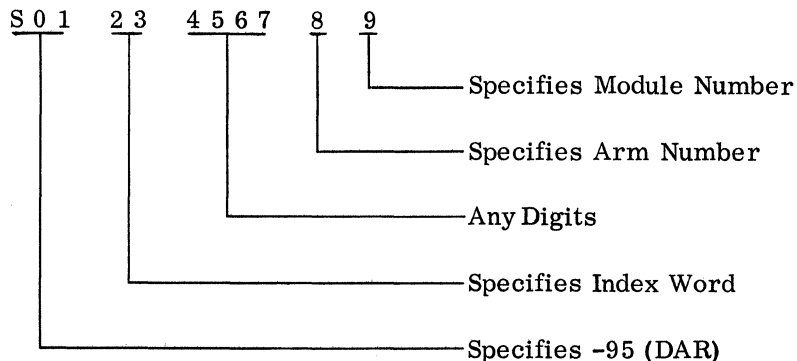
<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Arith Ser RI Drv-3	DS Insert Ctrl, Find Free Acc Ctin, C5	5D-44 05.31
Arith Pos 9 RO	F 1-9, AP	4C-44.03.01
Arith Pos 8 RO	F 9-8, AP	4H-44.03.01
Arith Pos 7 RO	F 8-7, AP	4C-44.03.02
Arith Pos 6 RO	F 7-6, AP	4H-44.03.02
Arith Pos 5 RO	F 6-5, AP	4C-44.03.03
Arith Pos 3 RO	F 4-3, AP	4C-44.03.04
Arith Par RO to AB	Find Free Acc Ctin, C 10-11	5C-44.05.54
A 3 AB RI Ctrl	Find Free Acc Ctin, C 10-11	5J-44.31.00
Develop DS Restart	DS Op, Matrix Intlk, C 11, Any DSUL On	3A-72.31.10
Set End Data Op	Find Free Acc Ctin DS Restart, CP	3F-42.40.41

5.14.04 Disk Arm Release (- 95)

The disk arm release operation is used to release the arm that was used during the DS read or write operation. The release arm operation can be performed in two ways:

1. The programmer can specify the arm that is to be reset.
2. The programmer can have the arm reset automatically when the system goes into a priority routine.

The format for the - 95 code is as follows:



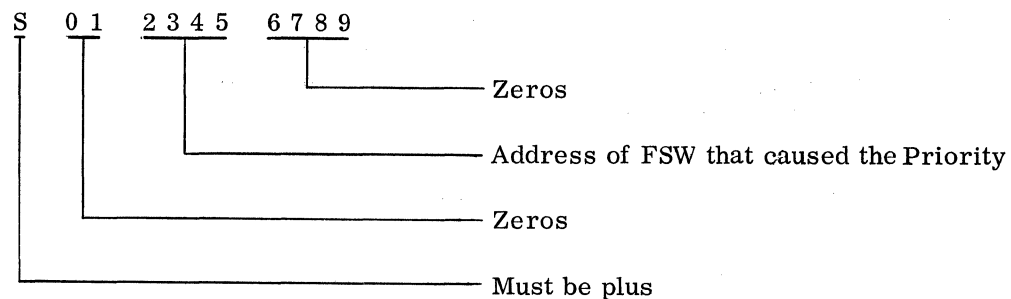
If the programmer wants to release the arm automatically, the disk read or write operation would have to be a + 9X operation (the + sign signifies a priority operation at the completion of a read or write operation).

At the completion of each disk read or write operation, a final status word is automatically created. This status word is then stored into its prescribed core storage location which is specified by the arm and unit that were used. The two low-order positions of each address (units and tens position) specifies the arm and unit.

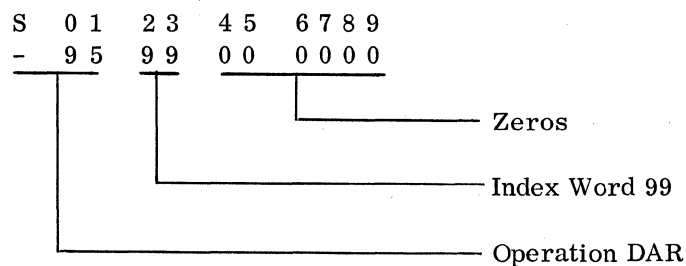
For example:

	BOX 0	BOX 1	BOX 2	BOX 3
ARM 0	0200	0201	0202	0203
ARM 1	0210	0211	0212	0213
ARM 2	0220	0221	0222	<u>0223</u>

Initiation of a disk storage priority routine, causes the address generator to automatically develop the address of the final status word. During the course of the priority routine, the address of the final status word is placed into index word 99. The format of index word 99 is as follows:



During the course of the priority routine, the programmer specifies a - 95 code. The format of the - 95 instruction would be as follows:



The index portion of the instruction specifies the index word that is to be added to the program register D. At the completion of the index add cycle, the contents of prog reg D are serially transferred to the addressed file unit to release the arm.

Sense Index Operation

During the Pre-sense of the instruction the - 9X code is sensed to turn on the no access latch. At the same time the index portion of the instruction is pre-sensed to specify an index operation.

The following MOP time the index request is developed to request the use of core storage. If memory can accept the request, it replies in the form of an Addr and Info Gate.

At addr gate time, the index ring is started and the index address is read out to the CAB. Zeros are inserted into the two high-order positions to validate the address (0099). At info gate time, the contents of index word 99 are read out to the information bus.

When the index ring reaches IX-5 time, the index word is read into the auxiliary register.

It can be seen from Figures 5.14-7 and 5.14-8 that when IX-4 time is reached the program ring is started. This furnishes the necessary timing to advance prog reg D and the auxiliary register to perform the index-add operation.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Index Latch	IX Tens Pos 6B IX Units Pos 6B, I Ctrl 4	4C-42.40.03
Perform Index Req	Index Latch, MOP	4E-42.40.03
Start Index Ring	A & P Addr Gt, Index Req	3B-42.41.05
RO IX Addr to CAB	IX Ctrl, Index Op	3A-42.41.20
Aux RI from IB	IX 3B, CP	5G-42.31.10
Start Program Ring	IX Ctrl 4 Info Gt	4C-41.12.01

Perform Index Add Operation

With the program ring running to furnish the necessary timing the index add operation is performed.

At C0-1 time both the auxiliary # 2 RO to channel # 2 and the program register D serial advance controls are brought up. At the same time, the true add control is brought up to inhibit the channel 1 and 2 zeros (Figure 5.14-8). The contents of prog reg D and Aux # 2 are read out to the adder.

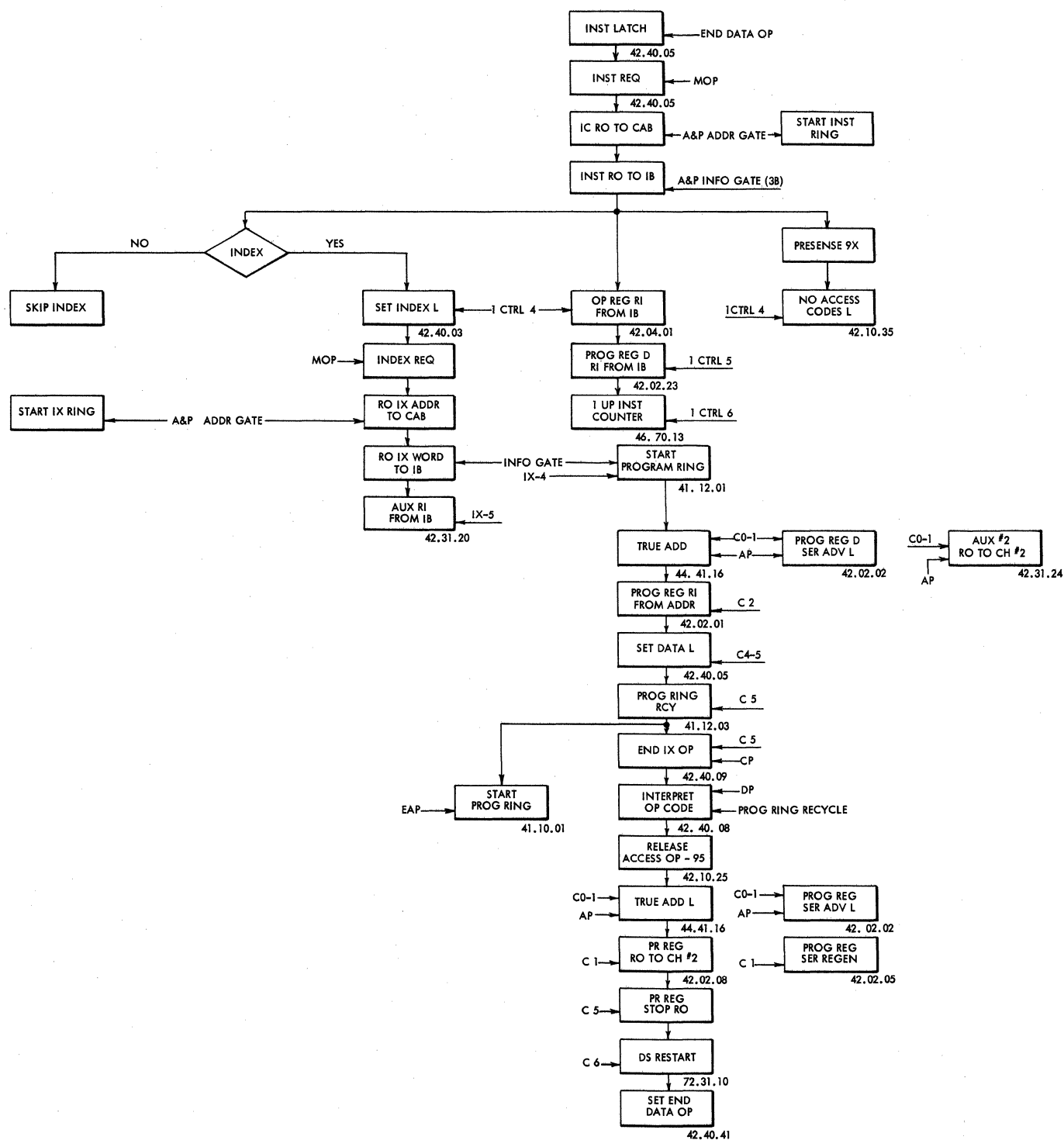


FIGURE 5.14-7. -95 DISK ARM RELEASE FLOW CHART

Due to the digit delay through the adder, the prog register RI from adder gate is turned on at C2 time.

At C4-5 time the Data latch is turned on. Due to the inhibit access latch being on, the data request is inhibited.

The end of the index-add operation is signalled at C5 time. Also at C5 time, a program ring recycle signal is initiated to restart the program ring. The program ring furnishes the necessary timing to transfer the contents of program register D to the addressed file unit to reset the arm.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Set Prog Reg D Ser Adv	Index Op, AP	4D-42.02.02
Aux # 2 RO to Ch # 2	Index Op, C0-1 AP	4C-42.31.24
Set True Add	Sign True Add, C0-1	3A-44.41.16
Prog Reg RI from Adder	Index Oper, C2	4D-42.02.01
Set Data Latch	Index Oper C4-5	2D-42.40.04
Develop Prog Ring Rcy	Index Oper, No Access Codes, C5	2H-41.12.03
End IX Operation	Index Op, C5, CP	4H-42.40.09

Transfer Program Register D to DS

With the program ring recycled, the Interpret Op Code signal is developed to initiate the release access operation.

At C0-1 time, the program-serial-advance latch is turned on to serially advance the program register onto channel 2 for validity-checking purposes. As the contents are read out to the channel 2 it is also sent to the file unit to release the arm. As the contents of the program register is read out it is serially regenerated.

At C1 time the units position of program register D is read out to the file units. This position contains the module number. The following digit time (C2 time) the contents of the tens position which contains the arm number is read out to reset the arm in the selected file unit. The remaining two digits of information are read out at C4 and C5 time. These digits have no effect on the operation.

Also at C5 time a Program Register Stop RO signal is developed to signify the end of the serial advance.

With the arm reset in the selected file unit, a disk-restart signal is developed at C6 time. This reset signal is sent to the A & P unit to develop the end-data op signal which signifies the end of the operation.

<u>Command</u>	<u>Timing</u>	<u>Logic and Location</u>
Develop Prog Ring Rcy	Index Op, No Access Codes, C5	2H-41.12.03
Set Interpret Op Code	Prog Ring Rcy, Data L, DP	3B-42.40.08
Set True Add	RLS Access, C0-1	4C-44.41.16
Set Prog Reg Ser Adv L	RLS Access, C0-1 AP	6B-42.02.02
PR Reg RO to Ch 2	RLS Access, C1	3E-42.02.08
Set PR Reg Ser Regen	RLS Access, C0-1, AP	5D-42.02.05
Stop PR Reg Ser RO	RLS Access, C5	6H-42.02.02
Develop DS Restart	RLS Access, C6	5B-75.60.11
Set End Data Op	RLS Access, DS Restart, CP	3F-42.40.41

5.15.00 START, STOP, AND RESET OPERATIONS

In normal operation the start, stop, and reset circuits are controlled by the console keys. In addition, several interval computer circuits are also used. The start and stop controls affect only the A & P area directly. The peripheral equipment (unit-record, tape, and disk storage) operate to the limit of their controls before stopping. In addition, they cannot start until instructed by the program. The reset controls affect the whole system and thus must only be used when the system is fully stopped.

5.15.01 Start Controls

The start control basically places the program into operation by causing the next instruction to read in. The instruction is executed, and operation is continued until a stop condition is encountered.

Console Start

A computer-start pulse originating at the console-start key serves as the normal start control. The pulse picks the start key latch at A-pulse time to provide proper sequence switching. The latch remains set until the input pulse falls. At C-pulse time the start latch is set to provide the pulse to start the program. The latch remains set for only 4 us. While the start latch is set, the start-interlock latch is set to prevent a second setting of the start latch. The start interlock latch remains set until the console stop key is depressed.

Forced Start (CE)

The forced-start pulse originates from the CE program reset key. It is gated to prevent its use during computer reset. The sequence of operation in setting the start key latch, the Start latch, and the start-interlock latch is the same as for a console start.

5.15.02 Stop Controls

The stop control discontinues program operation. It may be initiated by normal depression of a key, by program control, or by detection of an error condition. The basic conditions are the same for any stop. The operation being performed is completed except in the case of either a clocking ~~or an address circuit~~ error. These ~~two~~ error types are such that the operation may destroy data if allowed to continue.

The stop signal is switched against the Instruction Control 3A pulse to set the stop latch. The latch remains set for only 4 us. At A-pulse the stop-interlock latch is set to inhibit a second set of the stop latch. The interlock remains set until the start circuit is impulsed. The output of the stop latch (CPU Stop) inhibits the read-in of the new instruction word and the up-dating of the instruction counter. The CPU stop pulse also initiates the display-on stop cycle to type out the instruction counter and the program register contents on the console typewriter. Following the typing, the system comes to a halt until the start key is again depressed.

Console Stop

A stop can be initiated from the console stop key at any time. The program-advance indicator goes out and stays out. The print out indicates the point reached in the program.

Program Stop

A stop can be initiated by the program by use of ± 00 Operation Code. The program-advance indicator flashes on and off to alert the operator. The console-stop key must be depressed before the program can be restarted with the start key. The type-out indicates the two addresses, and the indicators show which address is to be used.

Address Stop

A stop is initiated when the address being transferred on the computer-address bus is equal to the address-stop-switch setting. The type-out shows the two possible addresses. Visual inspection determines if the instruction or data address caused the stop (the instruction address is plus one). The program-advance indicator flashes on and off with the stop. The stop key must be depressed before restart.

Single Cycle Stop

When the console switches are set for single cycle, a stop is initiated at the end of each Instruction. The system stops with a type-out for program inspection. The program advance indicator flashes. The stop key must be depressed prior to each restart.

Error Stops

Most of the error conditions occurring within the A & P area are mixed to set the any-error latch. Under normal operation this produces a system stop in much the same manner as a console stop. The program is stopped at the start of the next instruction. If a priority-interrupt is initiated at this point, it is suspended until after the restart. In either case an automatic display of the instruction counter, and program register is forced on the console. This type of error stop is referred to as a Soft Stop.

Two error conditions, which could cause loss of data, force an Immediate or Hard Stop. The stop latch is set without waiting for the end of the operation. An automatic type-out is initiated, as with any stop, but may fail because of the error. The two errors falling into this class are clocking and ~~address circuits errors~~.

The type-out feature indicates the point in the program where the error occurred. The error signals are divided into three classes to set indicator lights on the console:

1. Those conditions, which result from a programming fault, light the program-check light.
2. Conditions resulting from machine control malfunction light the arithmetic-check light.
3. An error in address or data transfer causes the validity-check light to turn on.

In normal operation the system stops, and the program-advance indicator flashes on and off to signal the operator. To restart, the stop key and the check-reset key must be depressed before the start key.

Most of the error conditions set individual latches for diagnostic purposes. These latches can be tested by use of the + 09 operation code to determine which error or errors caused the stop. For CE tests the machine-stop feature may be suspended, and the error latches tested and reset with the diagnostic code.

The following is a brief discussion of the purpose and effects of the various error signals. The signals are listed in order by their diagnostic test number.

Carry-No Carry (Logic 44.40.22)

The carry-no carry error signal sets diagnostic latch 1 and mixes to light the arithmetic-check indicator. Each adder operation produces, in addition to the sum digit, either a Carry or a No-Carry signal setting its own output latch. These signals are used to gate the entry for the next digit. An error signal is developed as the result of either setting no latch or setting both latches. An additional adder cycle will cause an adder-output-validity error for either condition and an adder-driver error if both latches were set.

Zero-Non Zero (Logic 44.40.20)

The Zero-Non Zero Error signal sets diagnostic latch 2 and mixes to light the arithmetic-check indicator. The zero-check latches are set with the zero latch on and the non-zero latch off at the start of the operation. The discontinuance of zeros causes the zero latch to turn off, while the presence of 0, 3, or 6 bits turns on the non-zero latch. An error signal is developed when both latches are set or both are reset. The presence of both signals also inhibits their output to prevent improper gating control. In some operations the lack of these signals causes additional errors.

Adder Driver (Logic 44.40.21)

The Adder Driver Error signal sets diagnostic latch 3 and mixes to light the arithmetic-check indicator. Normal adder operation requires one horizontal and one vertical drive line be driven. Extra drive lines in either direction causes more than one core to be set. A special core in each plane is wound with all of the drive lines. When more than one line is driven the core is set in the same manner as for a digit. The sense output of these cores becomes the error signal. This condition can occur from a Carry-No Carry Error, a True-Complement Error, or an invalid input. The adder-output-validity may also indicate an error.

Adder Output Validity (Logic 42.51.01)

The Adder Output Validity Error signal sets Diagnostic Latch 4 and mixes to light the validity-check indicator. The adder-output must always have a valid digit combination. When the adder is not in use, zeros are inserted during each digit time. The error signal is developed when less than two or greater than two bits are detected (Figure 2.3-7).

Channel 1 Validity (Logic 42.50.01)

The Channel 1 Validity Error signal sets Diagnostic Latch 5 and mixes to light the validity-check indicator. The channel is checked each digit time for a valid digit combination. Zeros are inserted when the channel is not in use. During adder operation the zeros are suppressed by the true or complement signal. The error signal is developed when less than two or more than two bits are detected (Figure 2.3-7). An error may also cause an Adder Output Validity Error or an Adder Driver Error.

Channel 2 Validity (Logic 42.50.02)

The Channel 2 Validity Error signal sets Diagnostic Latch 6 and mixes to light the validity-check indicator. The channel is checked each digit time for a valid digit combination. Zeros are inserted when the channel is not in use. During adder operation the zeros are suppressed by the true or complement signal. The error signal is developed when less than two or more than two bits are detected (Figure 2.3-7). An error may also cause an Adder Output Validity Error or an Adder Driver Error.

A & P Match (Logic 42.14.05)

The A & P Match Error signal sets Diagnostic Latch 7 and mixes to light the arithmetic check indicator. The match-mismatch circuit (Figure 2.3-8) has two latches, one set and one reset, for a match condition. The latches are reversed for a mismatch by separate circuits. Extra and missing bit conditions cause the latches to change separately. The error signal is developed if both latches are on or both are off.

Operation Sign (Alpha) (Logic 42.11.01)

The Operation Sign Error signal sets Diagnostic Latch 8 and mixes to light the program check indicator. An alpha for an instruction cannot be interpreted. An error signal is developed from the presence of the Alpha sign bits.

Op - 03 PR 4 Instruction (Logic 42.06.20)

The Operation - 03 PR 4 Instruction Error signal sets Diagnostic Latch 9 and mixes to light the program check indicator. The - 03 code is a sign set instruction. If PR 4 position of the instruction word is other than a sign digit (3, 6, or 9) the error signal is developed.

Branch-No Branch (Logic 42.40.06)

The Branch-No Branch Error signal sets Diagnostic Latch 10 and mixes to light the arithmetic check indicator. In normal operation either the branch latch or the no-branch latch must be set for each operation. Operations in which branching is not required cause the no-branch latch to be forced. An error signal is developed if the operation results in both latches off or both latches on.

Index-Skip Index (Logic 42.40.03)

The Index-Skip Index Error signal sets Diagnostic Latch 11 and mixes to light the arithmetic check indicator. In normal operation either the index latch or the skip-index latch is set for each instruction. An error signal is developed if both are set. No check is made for the condition of neither latch set because this is the normal machine stop condition.

HI-LO-Equal Failure (Logic 45.00.04)

The High-Low-Equal Failure signal sets Diagnostic Latch 12 and is mixed to light the arithmetic check indicator. One of the high-low-equal latches should be set for a compare type of operation. An error signal is developed if no latches are set or if two or more latches are set. If an error exists the latch reset is placed under control of the error-reset control.

Set-Reset (Logic 44.42.05)

The Set-Reset Error signal sets Diagnostic Latch 13 and mixes to light the arithmetic check indicator. An error signal is developed if an overflow condition fails to set its latch or if the condition latch fails to reset when tested. Refer to sections 5.8.07 and 5.8.09 for discussion on Branch if Overflow in Accumulator # and Field Overflow Control, respectively.

One-Up Error (Logic 42.13.02)

The One-Up Error signal sets Diagnostic Latch 14 and mixes to light the arithmetic check indicator. The one-up circuit functions both as a one-up and zero-up adder. Separate latches are set for the two conditions. Only one latch should be set for correct operation. An error signal is developed if both latches are set.

TLU Transfer Control (Logic 45.08.10)

The Table Lookup and Transfer Control Error signal sets Diagnostic Latch 15 and mixes to light the program check indicator. The record control word system is used in several block transfer operations. Several conditions must be checked to insure that an area of core storage is not selected in error. An error signal is developed from the following conditions:

1. The increment value for table lookup, stored in positions 6 through 9 in word 0098, is zero.
2. The record control word start address is greater than the stop address and thus does not define an area.
3. On numeric-to-alpha conversions, if the start address is equal to the stop, the necessary pair of words cannot be obtained.
4. The transfer-match latch fails to set, when the A & P match latch indicates a start-and-stop match.
5. An alpha sign on the record control word sets its own error, but is also mixed to force a match condition to end the operation.

Table Look-Up (Logic 45.09.14)

The Table Lookup Error signal sets Diagnostic Latch 16 and mixes to light the program check indicator. The Table Lookup Lowest operation requires special controls to end the operation because the entire core storage area is searched. The Memory Latch 1 is set for this control. If the latch fails to set, an error signal is developed to indicate the failure.

Out of Range Shift (Logic 42.11.28)

The Out-of-Range Shift Error signal sets Diagnostic Latch 17 and mixes to light the program check indicator. In normal shift operations the controls are arranged to handle shifts within the register. An error signal is developed if the specified shift is greater than register size (11 up for single and 21 up for double shift). The error signal also forces an end of operation.

Record Control Word Sign Alpha (Logic 45.10.02)

The Record Control Word Sign Alpha Error signal sets Diagnostic Latch 18 and mixes to light the program check indicator. The record control word sign is used to indicate the status of the core storage area. An alpha sign has no designation or control in the operation. An error signal is developed if an alpha sign occurs. The error signal also feeds the tabel lookup and transfer control error to end the operation.

Significant Digit Scanner (Logic 44.06.07)

The Significant Digit Scanner Error signal sets Diagnostic Latch 19 and mixes to light the arithmetic check indicator. In normal add operations using field control, the significant digit scanner is used to determine the number of digits in the accumulator factor. To insure that no digits are left after the transfer, the contents of the auxiliary register are read out to the scanner. A zero-sense output resets the SDS check latch. An error signal is developed when:

1. The check latch fails to reset by the end of the program cycle ring.
2. Any significant digit reading out to the Auxiliary 3 Output Latches while the A-test latch is set.

Index Operation Sign Alpha (Logic 44.35.66)

The Index Operation Sign Alpha Error signal sets Diagnostic Latch 21 and mixes to light the program check indicator. Index operations can only be performed with numeric signs. No provision is made to justify an alpha sign. An error signal is developed if the Operated Alpha Sign latch is set on an index operation.

Core Storage Sign Reverse (Logic 44.35.63)

The Memory Sign Reverse Error signal sets Diagnostic Latch 22 and is used to light the sign-change indicator. A memory sign reverse is considered an error when the sign-sense latch is off. The basic error signal is developed when the core storage sign is changed unless permitted by the operation:

1. A store operation under field control sets the sign to that of the accumulator.
2. An add to storage operation with an alpha accumulator sign.
3. An add-to-storage operation with a complement adjust.

The sense feature allows the program to continue with provisions to test the latch and branch.

True-Complement (Logic 44.41.15)

The True-Complement Error signal sets Diagnostic Latch 23 and mixes to light the arithmetic check indicator. Each adder operation requires the set of either the true-add or the complement-add latch. They serve to gate the B entry and inhibit channel validity zeros. An error signal is developed if both latches are set at the same time. The condition of no latches set is not checked because of the gating problem. The adder-output-validity error will show with either both on or both off, while the adder-driver error will show only with both set.

Divide Overflow (Logic 45.60.03)

The Divide Overflow Error signal sets Diagnostic Latch 24 and mixes to light the program check indicator. In a divide operation the factors must be placed in such a position as to allow a maximum quotient digit of nine to be developed. This means that the absolute value of the divisor (A3) is greater than the absolute value of the Accumulator 1 portion of the dividend. An error signal is developed if the shift register reads out to the A-First latch and the reduction continues. The signal also sets the End Divide latch to end the operation.

Arithmetic Register Zero Insert (Logic 44.02.32)

The Arithmetic Register Zero Insert Error signal sets Diagnostic Latch 25 and mixes to light the arithmetic check indicator. The individual position zero latches for the arithmetic register are set on at the start of an add cycle. As each digit time passes, a latch is reset starting with the low order. The high-order latches remaining set at the end of the operation are used to insert zeros into their respective positions. An error signal is developed if any latch fails to reset at the proper time or any latch sets or resets in error (Figure 5.15-1).

Field Ring Error (Logic 44.02.24)

The Field Ring Error signal sets Diagnostic Latch 26 and mixes to light the arithmetic check indicator. In field control applications of the field ring a check feature using the field-size register determines correct operation. If the field length signal coincides with field ring last, the check latch is reset. An error signal is developed if the check latch is still on at field-ring test time, or if the check latch fails to reset by force on full field.

Field Program Error (Logic 44.02.23)

The Field Program Error signal sets Diagnostic Latch 27 and mixes to light the program check indicator. A successful field control can be effected only if the Position 5 digit is equal to or greater than Position 4 of the instruction. An error signal is developed if an adder-carry occurs during the determination of the field size digit. In most operations a field-ring Run-out will also occur.

Field Ring Run-Out (Logic 44.03.06)

The Field Ring Run-Out Error signal sets Diagnostic Latch 28 and mixes to light the arithmetic check indicator. The field ring normally runs between the start and stop positions defined by the program register field. An error signal is developed if the ring has reached the last position (FX) at any time other than field ring test. This condition can occur with a field program error.

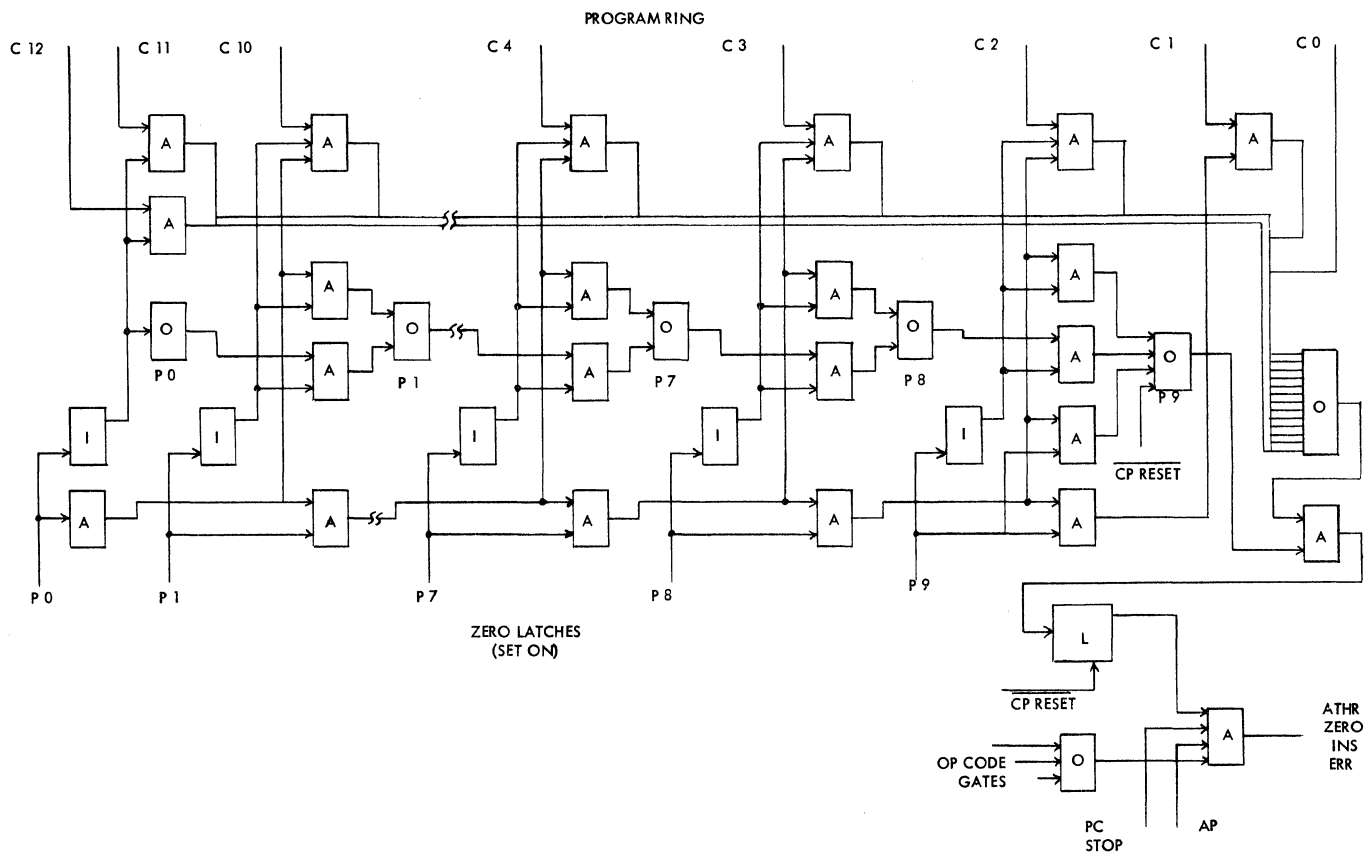


FIGURE 5.15-1. ARITHMETIC REGISTER ZERO INSERT ERROR CHECK

Program Ring Run-Out (Logic 41.10.07)

The Program Ring Run-Out Error signal sets Diagnostic Latch 29 and mixes to light the arithmetic check indicator. The program ring is stopped at the end of each operation. It has enough positions to perform the longest normal operations. A few cases use multiple cycles with ring recycle. An error signal is developed if the program ring reaches the last position (C 12) without the ring-stop latch being turned on.

Record Control Word (Logic 45.45.36)

The Record Control Word Error signal sets Diagnostic Latch 30 and mixes to light the program check indicator. The record control word for card transfer uses separate control switching from other transfer operations. A separate error signal is developed from this switching. This occurs when the start address is greater than the stop address and thus does not define an area within core storage.

Tape Control (Logic 46.70.15)

The Tape Control Error signal sets Diagnostic Latch 31 and mixes to light the program check indicator. In tape operation it is frequently desirable to operate with the priority controls masked. If an error occurs in a tape operation, it is not processed under this condition. Use of the same tape drive again would destroy the final status word. An error signal is developed if the drive is requested when the priority signal is masked or if the request is made in interrupt mode with a priority waiting.

Information Bus Validity (Logic 47.00.36)

The Information Bus Validity Error signal sets Diagnostic Latch 32 and mixes to light the validity-check indicator. The information bus must have valid digits during any word transfer. This is checked in A & P with the core-validity-check circuit (Figure 2.3-7). The core storage entry is further checked for valid digits by testing the inhibit driver combination. The information-bus validity-check signal is sent to the core control area. It is mixed with the inhibit-driver validity-check signal to form the IB-Inhibit Validity Check. This signal is switched with the A & P information gate to provide IB-Inhibit Validity Check A & P to the A & P area. The signal is further switched with card control signals to develop the IB Validity Error.

Address Validity Check A & P (Logic 55.04.05)

The A & P Address Validity Check Error signal sets Diagnostic Latch 33 and mixes to light the validity-check indicator. The basic address validity check is developed in the core control area. If an error is detected, the memory cycle is inhibited. The error signal is switched with the address gate to determine the area affected. The validity check is not effective on transfers on the computer address bus within the A & P area.

Instruction Word Error (Logic 42.11.00)

The Instruction Word Error signal sets Diagnostic Latch 35 and mixes to light the program check indicator. Instructions which require the use of the program register field to locate a factor, cannot use word 0000. An error signal is developed if both digits are zero on these operation codes.

Stacking Error (Logic 48.95.12)

The Stacking Error signal sets Diagnostic Latch 36 and mixes to light the arithmetic check indicator. A check is made on the stacking latch set and reset instructions to determine that the operation was performed. In normal operation the latches are set from the synchronizer, and a reply signal returns when the latch is set to release the synchronizer. An error signal is developed on the set-reset operation by switching the test signals with the operation gate. A transfer to D indicates the latch is set. The branch latch is not set.

Overflow Stop (Logic 44.42.05)

The overflow-stop signal does not set a diagnostic latch. Separate indicators are used for each of the accumulators and for the core storage field. The respective overflow latches set if a carry develops in the high-order position. Console switches determine if the program should be stopped. The indicators light when the latches are set for either a sense or stop operation. The overflow signal is treated as an error signal when the console switch is set to Stop. Depression of the error-reset key resets the latch.

Clocking Error (Logic 31.03.04)

The clocking-error signal is not mixed in the any-error signal and no diagnostic latch is set. The error forces an immediate stop with the clocking check indicator lighted. The error signal is developed when any of the clocking rings in the basic system are out of step. The clocks are resynchronized by use of the computer-reset key.

Address Circuit Error

The address-circuit-error signal is not mixed in the any-error signal, and no diagnostic latch is set. A validity-check indication is made on the console. The error signal forces an immediate stop. The basic error signal is developed in the core storage area from an invalid "X" and "Y" drive line combination. A latch is set in the core control area to inhibit the development of any address gates. The error is sensed in the A & P area by switching with A & P memory request.

5.15.03 Reset Controls

The reset controls normally originate from the tree console reset keys. Exceptions occur during a power-on sequence and during CE panel operation. The distribution switching is performed in the 7600 input/output control box, when the initial timing and mixing occur (Figure 5.15-2). The output signals are repeated during each drum revolution until the key is released. The following conditions are fed to the 7601 for reset operations:

1. The computer-reset key sets the computer-reset latch to supply a timed computer-reset signal.
2. The program-reset key sets the program-reset latch, which, in turn, forces the computer-reset latch. Timed signals for both computer reset and program reset are supplied.
3. The (check) reset key sets the (error) reset latch, which, in turn, sets the program-reset latch and the computer-reset latch. Timed signals for (error) reset, program reset, and computer reset are all supplied.
4. The power-on sequence from the console supplies the initial computer reset latch for a timed reset signal.

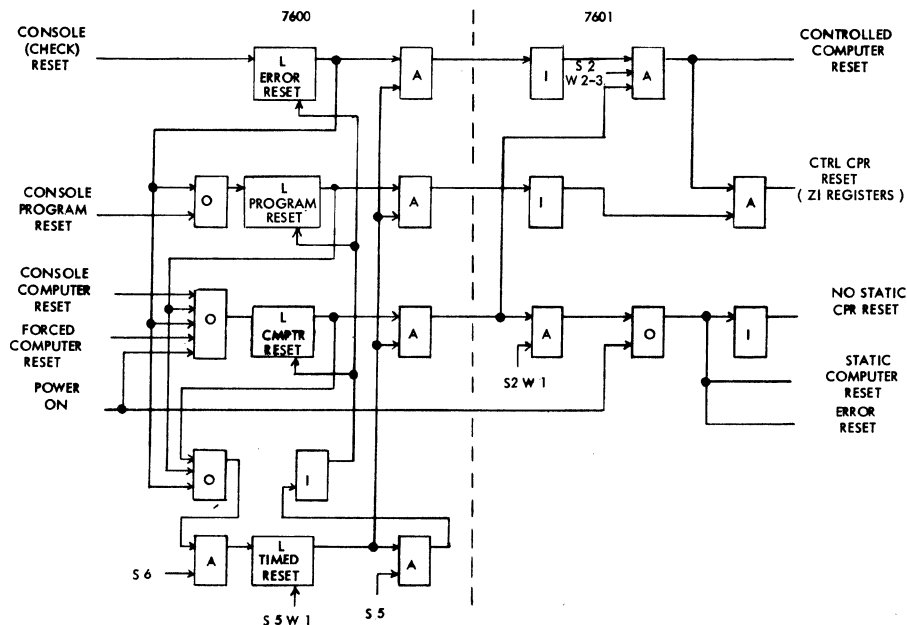


FIGURE 5.15-2. SIMPLIFIED LOGIC OF RESET SWITCHING

5. The forced-computer reset originating at the CE panel is used to force a computer reset in the normal manner.

In the 7601 boxes, four basic reset lines are developed by switching the input signals. Figure 5.15-2 shows this development in simple logic.

1. Error Reset, timed during each Drum Sector 2 Word 1, is developed from the computer-reset signal. It is used to reset error conditions and diagnostic latches.
2. Static Computer Reset, timed during each Drum Sector 2 Word 1, is developed from the computer-reset signal. This line is developed from the same switching as the error reset, but normally exists in the inverted phase. It is used to reset the various condition, cycling, testing, and gating latches, which do not have periodic resets. Some of the condition latch resets are inhibited in the distribution by the program reset line.
3. Controlled Computer Reset, timed during each Drum Sector 2 Word 2 and 3, is developed from the computer reset signal. It is inhibited entirely by the presence of the (error) reset signal and in part by the presence of the program reset signal. It is used to zero insert magnetic-core registers in a time sequence. The program ring is operated during the Word 2 time. A few latches, which require a late computer reset are also connected. The timing chart shows the timing sequence for Sector 2 Word 2 of the reset (Figure 5.15-3).
4. Program Reset, which is timed only in the 7600, has no further switching of the program-reset signal. It is used only to inhibit portions of the computer-reset controls, which are not required for the intermediate level reset.

(Check) Reset Circuit

Depression of the (check) reset key effects, what can be considered, a minor reset. It is normally used when the system stops with one of the processing check indicators lighted. The following list covers the general classification of latches and circuits being reset:

1. Reset program branch.
2. Reset validity-error latches and circuits.
3. Reset arithmetic-error latches and circuits.
4. Reset program-error latches and circuits.
5. Reset cycle-and-gate latches without periodic resets.
6. Reset sign-and-overflow errors set to stop.
7. Reset Hi-Eq-Lo latches on failure and extra sets.

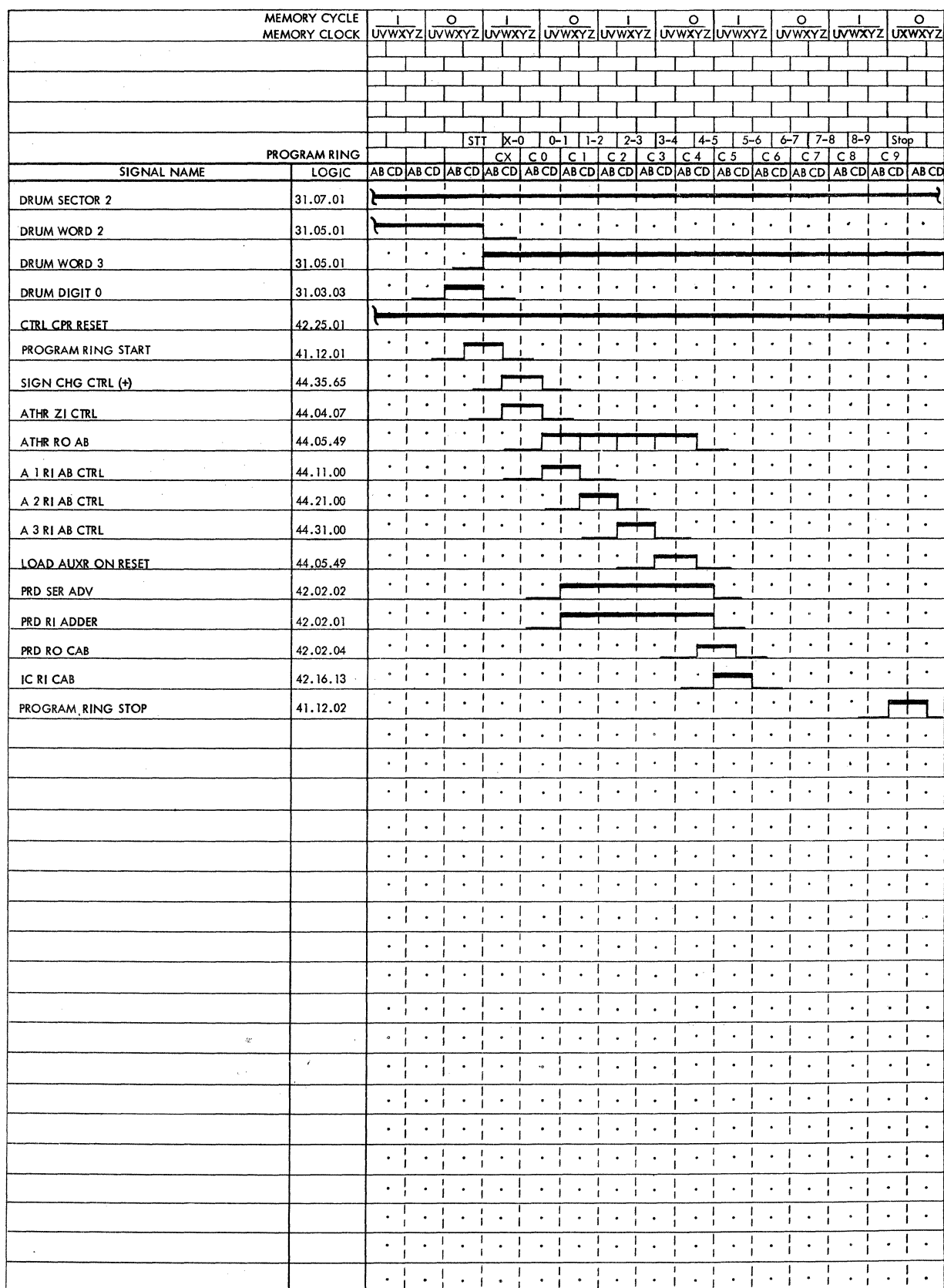


FIGURE 5.15-3. CONTROLLED COMPUTER RESET TIMING CHART

Program Reset Circuit

Depression of the program-reset key effects, what can be considered, an intermediate reset. It is normally used when it is discovered that the program has reached a point of incorrect operation or an error condition has developed. The following list covers the general classification of latches and circuits being reset:

1. All conditions reset by (check) reset circuit (above).
2. Reset program register D to 0000.
3. Reset instruction.
4. Reset A & P match latch.

Computer Reset Circuit

Depression of the computer-reset key effects a full or major reset. It is normally used when peripheral equipment is in error, the clocking system is in error, or a new program is being started. The following list covers the general classification of latches and circuits being reset:

1. All conditions reset by (check) reset circuit (above).
2. All conditions reset by program reset circuit (above).
3. Reset clocking error and clocks.
4. Reset priority mode.
5. Reset priority mask latches.
6. Reset priority stacking latches (7602).
7. Set field sense/stop latches to stop.
8. Reset indicator condition latches.
9. Set arithmetic and auxiliary register to zero.
10. Set accumulators to plus zeros.
11. Set program register to plus zeros.